

**APLIKASI TEKNIK PENCARIAN CERDAS
MENU KULINER
BERBASIS ANDROID**



SKRIPSI

Diajukan untuk memenuhi salah satu syarat guna mencapai gelar
Sarjana Komputer pada Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Alauddin Makassar

Oleh :

LAYLI LATIFAH LESTARY

NIM : 60200107002

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) ALAUDDIN
MAKASSAR
2013**

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Layli Latifah Lestary

NIM : 60200107002

Jurusan : Teknik Informatika

Judul Skripsi : Aplikasi Teknik Pencarian Cerdas Menu Kuliner Berbasis
Android.

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar merupakan hasil karya saya sendiri dan bukan merupakan pengambil alihan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut sesuai ketentuan yang berlaku.

Makassar, 26 Agustus 2013

Penulis,

Layli Latifah Lestary
NIM : 60200107002

PERSETUJUAN PEMBIMBING

Pembimbing penulisan skripsi saudara **Layli Latifah Lestary**, dengan NIM : **60200107002**, mahasiswa Jurusan Teknik Informatika pada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar, setelah dengan seksama meneliti dan mengoreksi skripsi yang bersangkutan dengan judul, *“Aplikasi Teknik Pencarian Cerdas Menu Kuliner Berbasis Android”*, memandang bahwa skripsi tersebut telah memenuhi syarat-syarat ilmiah dan dapat disetujui untuk diajukan ke sidang *Munaqasyah*.

Demikian persetujuan ini diberikan untuk proses selanjutnya.

Makassar, 26 Agustus 2013

Pembimbing I

Pembimbing II

Mustikasari, S. Kom., M. Kom

Nur Afif, S.T.,M.T

PENGESAHAN SKRIPSI

Skripsi yang berjudul “**Aplikasi Teknik Pencarian Cerdas Menu Kuliner Berbasis Android**” yang disusun oleh Saudari Layli Latifah Lestary dengan NIM : 60200107002, Mahasiswi Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Alauddin Makassar, telah diuji dan dipertahankan dalam sidang Munaqasyah yang diselenggarakan pada hari Rabu, 4 September 2013 M dinyatakan telah dapat diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam Jurusan Teknik Informatika dengan beberapa perbaikan.

Makassar, 4 September 2013 M
28 Syawal 1434 H

DEWAN PENGUJI

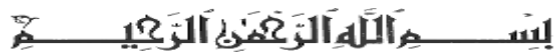
- | | | |
|------------------|--------------------------------------|---------|
| 1. Ketua | : Dr. Muhammad Halifah Mustami, M.Pd | (.....) |
| 2. Sekretaris | : Faisal, S.T., M.T | (.....) |
| 3. Munaqisy I | : Mega Orina Fitri, S.T., M.T | (.....) |
| 4. Munaqisy II | : Yusran Bobihu, S.Kom., M.Si | (.....) |
| 5. Munaqisy III | : Hasyim Haddade, S.Ag., M.Ag | (.....) |
| 6. Pembimbing I | : Mustikasari, S.Kom., M.Kom | (.....) |
| 7. Pembimbing II | : Nur Afif, S.T., M.T | (.....) |

Diketahui oleh :

**Dekan Fakultas Sains dan Teknologi
UIN Alauddin Makassar**

Dr. Muhammad Khalifah Mustami, M.Pd
NIP. 19710412 200003 1 001

KATA PENGANTAR



Assalamu'alaikum Wr.Wb

Puji syukur penulis panjatkan kehadiran Allah SWT, atas limpahan rahmat dan hidayah-Nya serta kesehatan yang tak terhingga, sehingga skripsi dengan judul “Aplikasi Teknik Pencarian Cerdas Menu Kuliner Berbasis Android” dapat terselesaikan. Shalawat dan salam tak lupa pula penulis haturkan kepada junjungan Nabi Muhammad SAW yang telah membawa kita semua dari alam kegelapan menuju alam terang benderang, dan dari dunia kebodohan menuju dunia yang penuh dengan ilmu pengetahuan.

Skripsi ini diajukan untuk memenuhi salah satu syarat dalam meraih gelar Sarjana Komputer pada Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Alauddin Makassar.

Penulis menyadari bahwa penyusunan skripsi ini tidak akan terselesaikan tepat waktu tanpa adanya bantuan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati penulis mengucapkan terima kasih yang sebesar-besarnya atas segala bantuan berupa dukungan, arahan, saran serta bantuan moril dan materil kepada penulis. Untuk itu penulis menghaturkan banyak terima kasih kepada :

1. Teristimewah kepada keluarga penulis, kedua orang tua, Drs. H. Ansar Djafar dan Dra. Hj. Hasniah Nonci yang telah merawat, membesarkan, membimbing, mendidik, dan mendo'akan penulis selama menempuh

pendidikan. Serta kedua saudara, Awal Muqsith, Lc., M.Phil dan Azhratul Jannah atas segala doa dan bantuan baik moril dan materil kepada penulis. Semoga Allah SWT senantiasa melimpahkan kesehatan dan kesejahteraan bagi mereka, Amin.

2. Bapak Prof. Dr. H. A. Qadir Gassing, H. T, MS. selaku Rektor Universitas Islam Negeri (UIN) Alauddin Makassar.
3. Bapak Dr. Muhammad Khalifah Mustami, M.Pd selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar.
4. Bapak Nur Afif, S.T., M.T selaku Ketua Jurusan Teknik Informatika serta Ibu Mega Orina Fitri, S.T., M.T selaku Sekretaris Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Alauddin Makassar.
5. Ibu Mustikasari, S.Kom., M.Kom dan Bapak Nur Afif, S.T., M.T selaku Pembimbing I dan Pembimbing II yang dengan keikhlasannya meluangkan waktu, tenaga dan pikirannya dalam pelaksanaan penelitian dan penyusunan skripsi.
6. Bapak Faisal, S.Kom., M.Kom, dan Bapak Yusran Bobihu, S.Kom., M.Si serta Bapak Ridwan A. Kambau, S.T., M.Kom yang senantiasa memberikan arahan kepada penulis dalam penyusunan skripsi.
7. Dosen-dosen yang mengajar di jurusan Teknik Informatika yang telah banyak memberikan asupan ilmu dari semester awal hingga semester akhir. Serta Staff Administrasi dan Akademik Fakultas Sains dan Teknologi yang banyak membantu dalam hal persuratan.

8. Spesial kepada sang suami tercinta dan tersayang Sutrisno, S.Kom beserta keluarga, yang telah memberikan sumbangsih dan selalu memberikan motivasi, semangat serta setia menemani sehingga penyusunan skripsi menjadi lebih indah.
9. Teman-teman seperjuangan Teknik Informatika Angkatan 2007 (ENIAC) yang telah menemani serta memberi canda tawa selama pendidikan dikampus.
10. Teman-teman tanpa pamrih khususnya Febry Sartika B, Dzunnah Wardanah dan Haidir Ali yang selalu gokil kala suntuk melanda.
11. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu, namun telah banyak terlibat membantu penulis dalam proses penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, maka dari itu dengan segala kerendahan hati penulis bersedia menerima kritik dan saran yang bersifat membangun dari semua pihak, baik dari pendidik maupun pembaca sehingga penulisan skripsi ini akan jauh lebih baik kedepannya.

Pada akhirnya penulis berkhidmat kepada Allah SWT, seraya memohon segala usaha hamba bernilai ibadah di sisi-Nya. Amin.

Fastabiqul Khairat, Wabillahi Taufiq Wal Hidayah

Wassalamu'alaikum Wr.Wb.

Makassar, 26 Agustus 2013

Penulis

DAFTAR ISI

	Hal
Halaman Judul	i
Halaman Pernyataan Keaslian Skripsi	ii
Persetujuan Pembimbing	iii
Halaman Pengesahan Skripsi	iv
Kata Pengantar	v
Daftar Isi	viii
Daftar Gambar	x
Daftar Tabel	xii
Abstrak	xiii
 BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Rumusan Masalah	6
C. Batasan Masalah	6
D. Pengertian Judul	6
E. Tujuan dan Manfaat Penelitian	8
 BAB II KAJIAN PUSTAKA	
A. Tinjauan Pustaka	9
B. Landasan Teori	10
 BAB III METODE PENELITIAN	
A. Lokasi Penelitian	50
B. Jenis Penelitian.....	50

	C. Metode Pengumpulan Data	50
	D. Hardware dan Software	51
	E. System Requirement	52
	F. Metode Pengujian Perangkat Lunak	52
BAB IV	ANALISIS DAN DESAIN SISTEM	
	A. Analisis Sistem	54
	B. Deskripsi Umum Perangkat Lunak	56
	C. Perancangan Sistem	58
BAB V	IMPLEMENTASI DAN PENGUJIAN SISTEM	
	A. Implementasi Perangkat Lunak	73
	B. Pengujian Sistem	73
BAB VI	KESIMPULAN DAN SARAN	
	A. Kesimpulan	81
	B. Saran	81
	Daftar Pustaka	82
	Lampiran	
	Riwayat Hidup Penulis	

DAFTAR GAMBAR

	Hal
Gambar II.1 Arsitektur Android	21
Gambar II.2 Hirarki View Android User Interface	30
Gambar IV.1 Use Case Diagram Libar Cook	59
Gambar IV.2 Class Diagram Libar Cook	60
Gambar IV.3 Activity Diagram Libar Cook	62
Gambar IV.4 Sequence Diagram.....	63
Gambar IV.5 Entitas ERD	64
Gambar IV.6 Attribut ERD	66
Gambar IV.7 Kardinalitas Relasi ERD	67
Gambar IV.8 ERD Libar Cook	68
Gambar IV.9 EER Libar Cook	69
Gambar IV.10 Diagram Rancangan Sistem.....	70
Gambar IV.11 Rancangan Jendela Utama.....	71
Gambar IV.12 Rancangan Jendela Daftar Bahan.....	71
Gambar IV.13 Rancangan Jendela Cari Resep.....	72
Gambar IV.14 Rancangan Jendela Tulis Resep.....	72
Gambar V.1 Jendela Halaman Utama	74
Gambar V.2 Jendela Daftar Bahan	75
Gambar V.3 Jendela Tambah Bahan	75
Gambar V.4 Jendela Tambah Bahan Berhasil	76
Gambar V.5 Jendela Cari Resep.....	77

Gambar V.6	Jendela Cari Resep Dengan Bahan Pangan	78
Gambar V.7	Jendela Hasil Pencarian Resep Dengan Bahan Pangan	78
Gambar V.8	Jendela Tulis Resep.....	79
Gambar V.9	Jendela Resep Favorit	80

DAFTAR TABEL

	Hal
Tabel II.1 Simbol Use Case Diagram.....	44
Tabel II.2 Simbol Sequence Diagram.....	47
Tabel II.3 Simbol Activity Diagram	48

ABSTRAK

Nama : Layli Latifah Lestary
NIM : 60200107002
Jurusan : Teknik Informatika
Judul : Aplikasi Teknik Pencarian Cerdas Menu Kuliner Berbasis Android

Manusia sebagai makhluk hidup tidak dapat terlepas dengan kebutuhan akan makanan. Salah satu aktifitas dalam membuat makanan adalah dengan memasak. Beragam makanan diolah dengan berbagai teknik memasak dan menggunakan rempah-rempah yang beragam pula. Indonesia sebagai negara yang kaya akan rempah-rempah sehingga membuat rakyatnya cerdas dalam mengolah makanan. Perkembangan teknologi memiliki peran penting dalam penyebaran resep-resep masakan khas Negara Indonesia, sehingga masakan-masakan dari Negara Indonesia tersebar sampai ke luar negeri. Berbagai macam resep-resep masakan tercipta sehingga membuat masyarakat tidak dapat mengetahui secara detail setiap resep masakan yang ada. Sehingga membuat masyarakat mulai mendokumentasikan resep-resep masakan dalam sebuah buku.

Aplikasi menu kuliner dibuat agar memudahkan masyarakat dalam mencari dan mengetahui resep-resep masakan. Sehingga proses masak memasak khususnya masyarakat yang baru belajar memasak tidak lagi terhambat karena kurangnya pengetahuan masyarakat tentang bahan dan cara membuat masakan. Aplikasi ini dapat diakses dengan menggunakan perangkat *mobile* berbasis Android. Penelitian ini merupakan penelitian kualitatif dengan metode pengumpulan data berupa *library research* dan *field research*.

Dari hasil penelitian ini diperoleh informasi bahwa aplikasi yang dirancang dapat membantu masyarakat dalam mengolah bahan pangan menjadi masakan. Dikarenakan aplikasi ini disertakan dengan kulkas *virtual* sebagai tempat penyimpanan bahan pangan layaknya kulkas sungguhan. Dengan penggunaan aplikasi ini diharapkan masyarakat dengan cepat dapat mencari dan mengetahui resep-resep masakan. Untuk pengembangan selanjutnya, diharapkan aplikasi menu kuliner ini ditambahkan dengan *fitur-fitur* yang belum ada dan *compatible* sehingga dapat berjalan disemua *platform* sistem operasi *mobile*.

Kata kunci : Aplikasi Kuliner, Resep Masakan, Rempah-rempah, Memasak, Sistem Operasi Android

BAB I

PENDAHULUAN

A. Latar Belakang

Manusia sebagai makhluk hidup tidak bisa lepas dengan kebutuhan akan makanan sebagai salah satu kebutuhan primer, karena untuk bertahan hidup manusia memerlukan tenaga yang diperoleh dari mengkonsumsi makanan. Unsur makanan terdiri dari beberapa jenis pertama makanan berat atau yang kita kenal sebagai lauk pauk yang dihidangkan biasanya bersama dengan nasi, kedua makanan ringan atau *cookies* yang biasanya dihidangkan sebagai makanan pencuci mulut atau bisa juga sebagai cemilan disaat waktu santai seperti kue.

Di Indonesia sendiri ada beragam jenis makanan dengan berbagai teknik dalam memasak. Keragaman masakan Indonesia tersebut merupakan pencerminan budaya dan tradisi yang berasal dari kepulauan Nusantara yang terdiri dari sekitar 6.000 pulau dengan beragam suku budaya. Indonesia secara umum dan hampir seluruh masakan Indonesia kaya dengan bumbu berasal dari rempah-rempah seperti kemiri, cabai, temu kunci, lengkuas, jahe, kencur, kunyit, kelapa dan gula aren dengan diikuti penggunaan teknik-teknik memasak menurut bahan dan tradisi-adat.

Dunia makanan atau yang lebih dikenal masyarakat dengan nama kuliner pada saat ini sedang digemari, dapat dilihat dari banyaknya bermunculan program-program acara televisi yang berunsur kuliner di berbagai stasiun televisi, mulai dari acara perlombaan masak-memasak, panduan memasak yang bisa mengajarkan cara memasak hingga wisata kuliner yang memberikan informasi masakan-masakan unik

yang belum pernah di ketahui. Fenomena ini menunjukkan *antusiasme* masyarakat terhadap kuliner saat ini sangat tinggi karena makanan sekarang ini tidak hanya sekedar untuk memenuhi kebutuhan gizi manusia semata tetapi telah menjadi suatu hobi yang dapat menghilangkan stress dan menanamkan kreativitas dalam memenuhi kebutuhan akan makanan yang baik dalam hal pemenuhan gizi dan kesehatan.

Firman Allah Swt. dalam surah Al- Baqarah ayat 168 :



Terjemahnya :

“Hai sekalian manusia, makanlah yang halal lagi baik dari apa yang terdapat di bumi, dan janganlah kamu mengikuti langkah-langkah syaitan; Karena Sesungguhnya syaitan itu adalah musuh yang nyata bagimu”. (Depag, 1971 : 41)

Pada ayat tersebut dijelaskan bahwa, Allah SWT menyerukan kepada seluruh umat manusia khususnya para muslim untuk mengkonsumsi makanan-makanan yang halal dan baik. Baik dalam artian makanan yang memiliki gizi dan vitamin yang baik untuk tubuh. Dan menjauhi segala makanan yang haram, dikarenakan makanan yang haram adalah makanan bagi para syaitan sedangkan syaitan adalah musuh yang nyata bagi orang-orang muslim.

Konsep makanan yang *halal* dan *thayyib* dalam Islam memiliki banyak makna dan pengertian dari para ulama. Raghīb al-Isfahani mengatakan bahwa kata *halal* secara *etimologi* berasal dari kata *halla-yahullu-hallan wa halalan wa hulalan* yang berarti melepaskan, menguraikan, membubarkan, memecahkan, membebaskan dan membolehkan. Sedangkan secara *terminologi*, kata *halal* mempunyai arti hal-hal yang boleh dan dapat dilakukan karena bebas atau tidak terikat dengan ketentuan-ketentuan yang melarangnya. Atau segala sesuatu yang bebas dari bahaya *duniawi* dan *ukhrawi*.

Kata *thayyib* menurut al-Isfahani menunjukkan sesuatu yang benar-benar baik. Bentuk jamak dari kata ini adalah *thayyibat* yang diambil dari derivasi *thabayathibu-thayyib-thayyibah* dengan beberapa makna, yaitu suci dan bersih, baik dan elok, enak dan halal. Untuk memenuhi kebutuhan primer hamba-Nya, Allah SWT dengan kasih sayang-Nya menganugerahkan bumi beserta isinya untuk dikelola dan dimanfaatkan dengan sebaik-baiknya oleh manusia. Kendati demikian, bukan berarti kita dapat memanfaatkan bumi beserta isinya itu dengan *mengeksplorasi* sebebas-bebasnya. Namun harus sesuai dengan apa yang digariskan *syariat*. Terkait dalam hal makanan dan minuman, tidak semua yang di bumi ini, baik binatang, tumbuhan maupun benda-benda lainnya itu halal dan baik (*thayyib*) bagi manusia. Ada yang memang dibolehkan (*halal*) dan ada yang dilarang (*haram*). Ada yang baik (*thayyib*), ada pula yang tidak baik (*khabits*).

Kemajuan teknologi yang terus berkembang menuju zaman yang modern berteknologi tinggi ini telah membawa dampak besar dalam kehidupan, karena dorongan kebutuhan hidup manusia yang terus bertambah . Dorongan kebutuhan tersebut menuntut manusia menciptakan inovasi baru yang diharapkan dapat meringankan segala aktifitas kehidupan manusia. Salah satu kemajuan teknologi yang berdampak besar saat ini adalah telepon genggam. Pada awal kemunculannya, telepon genggam dikenal hanya sebagai alat komunikasi semata, namun sekarang ini telepon genggam bukan hanya sekedar alat komunikasi semata tetapi mempunyai banyak fitur selain untuk berkomunikasi setelah mengalami proses pengembangan telepon genggam. Sekarang telepon genggam telah menggunakan unsur-unsur komputer baik itu dari segi perangkat lunak dan juga perangkat keras atau dapat dikatakan telepon genggam saat ini telah menjadi mini komputer yang dapat dibawa kemana-mana secara praktis (Gunadi, 2008 : 7).

Telepon genggam saat ini telah menjadi kebutuhan primer bagi manusia hampir setiap orang memiliki telepon genggam untuk membantu mereka berkomunikasi. Namun sekarang telepon genggam tidak hanya berfungsi untuk berkomunikasi semata tapi juga sebagai sarana hiburan dengan dilengkapinya aplikasi-aplikasi didalamnya yang bersifat menghibur, selain itu juga telepon genggam yang telah dikategorikan sebagai telepon genggam cerdas atau sering kita dengar dengan sebutan *smartphone* telah dilengkapi aplikasi-aplikasi cerdas yang membantu kita dalam memecahkan permasalahan di kehidupan sehari - hari seperti penunjuk arah , pembelajaran dan masih banyak lagi.

Perkembangan *smartphone* tidak hanya pada sisi aplikasinya saja tetapi juga pada sisi media penyimpanan dan fungsinya. Pada *smartphone* media penyimpanan data semakin bertambah, memungkinkan *smartphone* dapat berfungsi sebagai alat komunikasi, penunjuk arah, peta, *library* serta buku dan catatan.

Terkait kebutuhan akan makanan dan komposisinya tentunya akan membuat manusia berpikir untuk mengatur pola hidup dan menginovasi menu masakan, yang dulunya harus berpedoman pada buku-buku resep sekarang tentunya dapat dipikirkan untuk mengadopsi menu masakan tersebut pada sebuah perangkat yang mudah dan selalu dapat ikut bersama kita setiap waktu.

Bisa dibayangkan seberapa banyak makanan yang tercipta oleh manusia di dunia ini dan seberapa banyak resep yang tidak mungkin dapat di ingat manusia satu persatu. Untuk itu penulis merancang suatu aplikasi yang memberikan informasi mengenai resep makanan yang diterapkan pada perangkat *mobile*. Dengan aplikasi ini diharapkan dapat memberikan kemudahan bagi masyarakat dalam mengakses kebutuhan informasi tersebut.

Berdasarkan dengan latar belakang masalah tersebut maka penulis membuat Tugas akhir ini dengan judul **“Aplikasi Teknik Pencarian Cerdas Menu Kuliner Berbasis Android”**

B. Rumusan Masalah

Dari latar belakang di atas, maka yang menjadi rumusan masalahnya adalah “bagaimana cara merancang aplikasi pencarian cerdas menu kuliner yang *user friendly* dan *simple* dalam membuat menu makanan untuk disajikan dalam kehidupan sehari-hari?”

C. Batasan Masalah

Adapun batasan masalahnya adalah :

1. Pembuatan aplikasi ini dikhususkan untuk mengetahui bahan makanan dan cara membuat beberapa masakan sehingga dapat langsung dipraktekkan dalam kehidupan sehari-hari.
2. Aplikasi yang dibuat berbasis *mobile* dan dapat berjalan pada perangkat dengan sistem operasi Android versi 2.2 (Froyo) sampai 4.2 (Jelly Bean).
3. Aplikasi ini hanya memuat resep – resep makanan yang halal menurut islam.
4. Aplikasi ini berjalan *stand alone* dan pada kondisi *offline*.

D. Pengertian Judul

Untuk menghindari salah pengertian judul ataupun makna yang terkandung pada judul skripsi ini, maka dijelaskan beberapa istilah dalam judul skripsi ini.

1. Aplikasi

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus) (Hendrayudi, 2010 : 143).

2. Teknik Pencarian

Teknik pencarian adalah metode yang digunakan dalam penelusuran suatu data sehingga dapat mempercepat proses penemuan suatu data oleh sistem guna mengefisienkan waktu (Herianto, 1993 : 346).

3. Cerdas

Cerdas adalah kemampuan untuk belajar memahami, memutuskan dan beropini berdasarkan alasan tertentu (Budiharto, 2007 : 26).

4. Menu

Menu adalah sebuah daftar makanan yang telah dilengkapi dengan harga masing-masing, yang disediakan dan ditampilkan untuk menarik pelanggan serta memberikan nilai terhadap sejumlah uang terhadap makanan yang ditawarkan (Ambarini, 2008 : 4).

5. Kuliner

Kuliner adalah suatu bagian hidup yang erat kaitannya dengan konsumsi makanan sehari-hari (Joyokusumo, 2009 : 3).

6. Android

Android adalah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi (Safaat, 2012 : 6).

E. Tujuan dan Manfaat Penelitian

1. Tujuan Penelitian

Untuk membuat aplikasi *mobile* yang dapat membantu masyarakat dalam mencari dan mengetahui resep masakan.

2. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah :

- a. Sistem dapat dijadikan sebagai media pembelajaran dalam melakukan *demo* masak memasak.
- b. Sistem ini dirancang khusus untuk membantu para masyarakat dalam mengetahui bahan dan tata cara membuat suatu masakan
- c. Sistem ini dapat dijadikan sebagai bahan referensi untuk mengetahui lebih dalam mengenai aplikasi menu kuliner.

BAB II

KAJIAN PUSTAKA

A. Tinjauan Pustaka

Banyak penelitian yang telah dilakukan untuk mengembangkan aplikasi yang berkaitan dengan menu masakan atau menu kuliner. Penelitian tersebut di antaranya adalah penelitian yang dikembangkan oleh Fizqy (2012). Aplikasi masakan yang dikembangkan meliputi masakan dari suatu daerah dan hanya berupa teks dari masakan tersebut, resep masakan berbentuk list dan sistem bekerja secara *offline*. Resep makanan yang tersedia hanya berasal dari masakan padang, kemudian di selaraskan dengan *database* resep masakan tersebut.

Penelitian lain yang berkaitan adalah penelitian yang dilakukan oleh Wahyu (2011). Pada penelitian ini dikembangkan aplikasi masakan kuliner yang terdiri dari enam daerah dan masing-masing daerah hanya memiliki lima macam resep. Sistem ini dibangun untuk digunakan pada *handset* android versi 2.2 Froyo. Sistem ini tidak menggunakan *database* untuk resep masakan, sehingga *user* tidak dapat menambahkan resep yang lain pada aplikasi ini.

Salah satu penelitian tentang kuliner yang dikembangkan oleh Eka (2010). Aplikasi ini mengembangkan pemesanan menu makanan pada sebuah restoran dengan menggunakan Visual Basic 6. Seluruh laporan berupa hasil penjualan makanan akan *direport* pada sistem dan kemudian *diprint out*. Aplikasi ini tidak ditemukan resep bahan makanan dan tata cara pembuatannya.

Hal serupa juga dikembangkan oleh Surya (2012) adalah sistem yang merekomendasikan menu makanan pada salah satu restoran dengan menggunakan metode *knowledge base recommendation*. Sistem ini diintegrasikan dengan *web* sehingga para konsumen dapat dengan cepat memesan makanan pada sebuah restoran. Apabila tidak terdapat koneksi internet, maka sistem ini tidak dapat digunakan. Pada pengembangan sistem ini hanya terdapat menu makanan tanpa ada resep dan cara membuat yang dilampirkan.

Melihat dari beberapa penelitian yang ada, belum ada yang membahas tentang pembuatan aplikasi teknik pencarian cerdas menu kuliner berbasis Android khususnya dengan menggunakan daftar bahan sebagai kunci dalam membuat sebuah resep masakan.

B. Landasan Teori

1. Memasak

Memasak adalah suatu proses menciptakan sebuah makanan dengan mengurutkan konsep yang ada yaitu langkah-langkah dalam memasak, yang berarti juga memasak adalah mencipta sebuah karya yang bisa kita nikmati dan juga orang lain. Makanan hasil olahan tersebut menjadi sesuatu yang mempunyai cita rasa tersendiri tergantung pada cara pengolahannya (Badiatul, 2009 : 6).

Cara memasak tertua dikenal sampai saat ini adalah membakar. Tidak ada yang tahu persis bagaimana manusia purba menemukan teknik ini, namun para ahli menduga bahwa mungkin manusia purba menemukannya ketika mereka memakan

daging hewan yang mati dalam kebakaran hutan. Teknik pembakaran dan pemanggangannya di atas bara api sendiri bisa ditemukan mulai dari masyarakat yang paling terisolir dan primitif sampai manusia modern. Kunci dalam memasuki fase variasi memasak yang lainnya, muncul ketika manusia menggunakan tanah liat untuk alat memasak (pot masak). Dengan penemuan tanah liat, dan juga dengan teknik memasak dengan batu, manusia menemukan cara-cara lain seperti merebus makanan, mengasap daging, sampai membuat oven tanah liat.

Seiring waktu, variasi memasak juga menjadi semakin maju dengan munculnya dan berkembangnya cara-cara bercocok tanam. Pada masa-masa ini, manusia juga belajar mengolah tumbuhan untuk pelengkap makanan, khususnya biji-bijian, yang kemudian menjadi sumber karbohidrat di meja makan.

2. Bahan Pangan

Bahan pangan atau disebut juga bahan pokok makanan adalah bahan-bahan yang dibutuhkan untuk melengkapi sebuah masakan. Bahan makanan berasal dari 2 jenis, yaitu bahan makanan yang bersumber dari hewan dan bahan makanan yang berasal dari tumbuhan. Bahan makanan yang berasal dari hewan disebut dengan bahan pangan hewani, sedangkan bahan makanan yang berasal dari tumbuhan disebut bahan pangan nabati (Cahana, 2006 : 3). .

Bahan pangan nabati adalah bahan -bahan makanan yang berasal dari tanaman (bisa berupa akar, batang, dahan, daun, bunga, buah atau beberapa bagian dari tanaman bahkan keseluruhannya) atau bahan makanan yang diolah dari bahan dasar dari tanaman.

Bahan pangan hewani merupakan bahan-bahan makanan yang berasal dari hewan atau olahan yang bahan dasarnya dari hasil hewan. Kedua bahan pangan ini memiliki karakteristik yang berbeda sehingga memerlukan penanganan dan pengolahan yang berbeda pula. Bahan pangan hewani meliputi susu, telur, daging dan ikan serta produk-produk olahannya yang bahan dasarnya berasal dari hasil hewani. (Cai, 2012).

3. Sistem Operasi Android

Pada tahun 2005 Google mengakuisisi Android Inc. yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Yang kemudian pada tahun itu juga memulai membangun platform Android secara intensif (Mulyadi, 2010 : 2).

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup system operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka (Safaat, 2011 : 1).

Operating System Android ini dikeluarkan oleh Google untuk menyaingi Windows, Apple dan pesaing-pesaingnya. Google sendiri mempunyai alasan cukup kuat untuk melirik pangsa seluler, karena perkembangan teknologi telepon seluler dewasa ini sudah bukan merupakan evolusi lagi, melainkan sebuah revolusi. Babak baru dalam dunia telekomunikasi nirkabel ini terus bergulir dengan cepat.

Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Secara garis besar, arsitektur Android terdiri atas *Applications* dan *Widgets*, *Applications Frameworks*, *Libraries*, *Android Run Time*, dan *Linux Kernel* (Supardi, 2011 : 4).

Pesatnya pertumbuhan Android yaitu platform yang sangat lengkap baik itu system operasinya, Aplikasi dan Tool Pengembangan, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas Open Source di dunia, sehingga android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

Salah satu hal yang membuat perangkat Android menarik adalah tersedianya banyak aplikasi yang bisa dipakai oleh penggunanya. Berdasarkan data terakhir pada saat event Google IO 2011, tercatat sudah ada 200.00 aplikasi yang tersedia di Android Market. Jumlah tersebut tidaklah sedikit.

4. The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari android adalah Dalvik Virtual Machine (DVM). Android berjalan di dalam Dalvik Virtual Machine (DVM) bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaannya dengan java virtual machine seperti Java ME (Java Mobile Edition), tetapi Android menggunakan Virtual

Machine sendiri dikustomisasi dan dirancang untuk memastikan bahwa beberapa *feature-feature* berjalan lebih efisien pada perangkat *mobile*.

Dalvik Virtual Machine (DVM) adalah “register bases” sementara Java Virtual Machine (JVM) adalah “stack bases”, DVM didesain dan ditulis oleh Dan Bornsten dan beberapa engineeres Goole lainnya. Jadi bisa kita katakan “Dalvik equals(Java) == False”. Dalvik Virtual Machine menggunakan kernel Linux untuk menangani fungsionalitas. tingkat rendah termasuk keamanan, threading dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C / C++ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari kernel linux yang digunakan dalam Android tersebut.

Semua hardware yang berbasis Android dijalankan dengan menggunakan Virtual Machine untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa Java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

5. Android SDK (*Software Development Kit*)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di *release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawahan *handphone/smartphone*.

Beberapa fitur-fitur android yang paling penting adalah :

- a. *Framework* Aplikasi yang mendukung penggantian *komponen* dan reusable.
- b. Mesin Virtual Dalvik dioptimalkan untuk perangkat mobile.
- c. Integrated *browser* berdasarkan *engine open source WebKit*.
- d. Grafis yang dioptimalkan dan didukung oleh *Libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1.0 (Opsional akselerasi hardware).
- e. SQLite

- f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
- g. Bluetooth, EDGE, 3G dan WiFi (tergantung hardware)
- h. Kamera, GPS, kompas, dan *accelerometer* (tergantung hardware)
- i. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, *profil* dan kinerja memori dan *plugins* untuk IDE Eclipse.

6. Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

a. Application dan Widgets

Application dan Widgets ini adalah *llyer* dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. Applications Frameworks

Android adalah “ *Open Development Platform*” yaitu Android menawarkan kepada pengembang atau member kemampuan kepada

pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notification*, dan sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali *komponen* yang sudah digunakan.

Jadi, *application framework* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Application frameworks* adalah sebagai berikut :

1. Satu set *Views* : yang dapat digunakan untuk membangun aplikasi meliputi list, grid, kotak teks, tombol, dan sebuah *embeddable web*.
2. *Content Provider* : yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain (misalnya kontak telepon), atau untuk membagi data tersebut.

3. *Resource Manager* : menyediakan akses ke kode non sumber lokal misalnya string, gambar, dan tata letak *file*.
4. *Notification Manager* : yang memungkinkan semua aplikasi menampilkan alert yang bisa dikustomisasi di dalam status bar.
5. *Activity Manager* : yang mengatur *life cycle* dari aplikasi dan menyediakan navigasi umum *backstack* (Mulyadi, 2011 : 8)

c. *Libraries*

Lapisan setelah Kernel Linux adalah *Libraries*. Android menyertakan satu set *library-library* dalam bahasa C/C++ yang digunakan oleh berbagai *komponen* yang ada pada sistem Android. Kemampuan ini dapat diakses oleh programmer melewati Android *application framework*. Sebagai contoh Android mendukung pemutaran format audio, video, dan gambar.

Berikut ini beberapa *core library* tersebut :

1. *System C library* : variasi dari implementasi BSD, berdasar pada sistem standar *C library* (libc) yang sesuai untuk perangkat berbasis Linux.
2. *Media Libraries* : Paket Video berdasarkan OpenCORE; *library* mendukung pemutaran rekaman dan banyak format populer audio dan video, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.

3. *Surface Manager* : mengatur akses ke subsistem layar, lapisan komposit 2D and grafis 3D dari berbagai aplikasi.
4. *LibWebCore* : mesin web browser modern yang powerfull.
5. *SGL* : mesin grafis 2D.
6. *3D Libraries* : implementasi berdasarkan OpenGL ES 1.0 APIs; *library* ini menggunakan *hardware 3D acceleration* dan *highly optimized 3D software rasterizer*.
7. *FreeType* : bitmap dan vektor font *rendering*.
8. *SQLite* : mesin *database* yang *powerful* dan ringan tersedia untuk semua aplikasi. (Reto, 2010 : 13)

d. Android Run Time

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi linux. Bertempat di *level* yang sama dengan *Libraries* adalah Android-Runtime. Android Runtime ini berisi *Core Libraries* dan Dalvik Virtual Machine.

Core Libraries mencakup serangkaian inti *library* Java, artinya Android menyertakan satu set *library-library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada *library-library* dasar bahasa pemrograman Java.

Dalvik adalah Java Virtual Machine yang memberi kekuatan pada sistem Android. Dalvik Virtual Machine ini di optimalkan untuk telepon seluler.

Setiap aplikasi yang berjalan pada Android berjalan pada prosesnya sendiri, dengan *instance* dari Dalvik Virtual Machine. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi Virtual Machine dengan efisien. Dalvik Virtual Machine dapat mengeksekusi *file* dengan format Dalvik Executable (.dex) yang telah dioptimasi untuk menggunakan memory seminimal mungkin. Virtual Machine ini *register-based*, dan menjalankan *class-class* yang dicompile menggunakan *compiler* Java yang kemudian ditransformasi menjadi format .dex menggunakan "dx" tool yang telah disertakan.

Dalvik Virtual Machine (DVM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti threading dan *low-level* memory management.

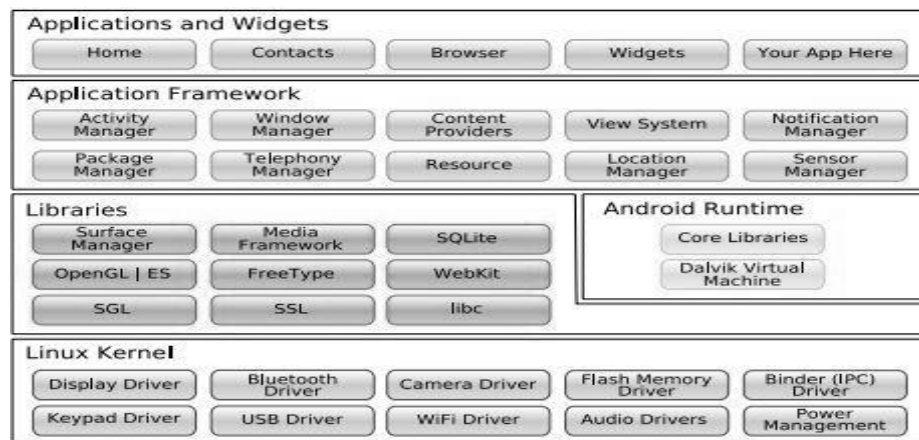
Perbedaan antara Dalvik VM dan Java murni adalah :

- 1) Dalvik VM menjalankan *file* .dex yang merupakan hasil konversi pada saat kompilasi program dari *.class* dan *.jar*. *File* .dex lebih padat dan efisien dari pada *file* *.class*. Hal ini akan menghemat daya dan mengurangi konsumsi memory.
- 2) Core Java *Libraries* yang ada di Android itu berbeda dengan Java SE maupun Java ME. (Ed Burnette, Hello Android, Introducing Google's Mobile Application

Development), edisi 3 (United States of America : Pragmatic Programmers LLC, 2010 : 33).

e. Linux Kernel

Linux Kernel adalah *llayer* dimana inti dari operating system dari Android itu berada. Berisi *file-file* system yang mengatur system processing, memory, *resource*, drivers, dan system-sistem operasi android lainnya. Linux Kernel yang digunakan android adalah linux kernel release 2.6



Gambar II.1. Arsitektur Android

Sumber : Nazruddin Sfaat, Android Pemrograman Aplikasi Mobile Smartphone dan Tablet Berbasis Android, 2011.

7. Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data *file* secara *resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di-package oleh tools yang dinamakan ‘apt tools’ ke dalam paket

android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang kita sebut dengan aplikasi dan nanti dapat diinstall di perangkat mobile.

Beberapa jenis *komponen* pada aplikasi Android, yaitu :

(a) *Activities*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan. Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *activity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan.

Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* Android menggunakan tumpukan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan

untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi, yaitu :

- 1) **Active**, setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.
- 2) **Paused**, dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukan dari pengguna. Dalam kasus ekstrim, Android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.
- 3) **Stopped**, ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi

kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive*.

- 4) **Inactive**, kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali. Kondisi transisi ini sepenuhnya ditangani oleh manajer memori Android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

(b) Service

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. *Komponen service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

(c) Broadcast Receiver

Broadcast Receiver merupakan *komponen* yang sebenarnya tidak melakukan apa-apa.

(d) Content Provider

Content Provider yaitu mekanisme yang memungkinkan sebuah aplikasi berbagi informasi dengan aplikasi lainnya. (Murphy, 2010 : 3).

8. Versi Android

Sejak diluncurkan pada tanggal 23 September 2008, Android telah mengalami beberapa kali pengembangan dan perbaikan terhadap versi sistem operasinya. Yang menarik dari versi-versi sistem operasi Android ini ialah penggunaan *codename* untuk setiap versi yang mengambil nama hidangan penutup dan memiliki pola keterurutan nama berdasarkan abjad.

Versi yang pertama adalah Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang diintegrasikan, CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*, kemampuan dial kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel), pengadaan resolusi VWGA.

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital zoom, dan Bluetooth 2.1.

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan Wi-Fi *Hotspot portable*, dan kemampuan *auto update* dalam aplikasi Android Market.

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi copy paste, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu (Safaat, 2011 : 11-12).

Android versi 3.0 (Honeycomb) diluncurkan pada Juli 2011. Tidak butuh waktu yang lama, Google kembali meluncurkan Android versi 4.0 yang diberi nama versi ICS atau Ice Cream Sandwich pada November 2011. Android versi 3.0 keatas adalah generasi *platform* yang digunakan untuk tablet pc. Sementara Android versi 4.0 ini sudah merupakan *platform* yang bisa dipakai si *smartphone* dan tablet pc. Pada tahun 2013, Google meluncurkan versi terbaru dari Android yaitu versi 4.1 dan 4.2 yang disebut dengan Android OS versi Jelly Bean.

9. User Interface

Android *User Interface* (UI) bisa dibuat dengan dua cara, yaitu dengan menggunakan kode XML atau menulisnya melalui kode Java. Membuatnya dengan XML sangat direkomendasikan, karena XML menggunakan konsep MVC (*Model-Viewer-Control*) yang memisahkan antara UI dengan kode logika program (*Logic Program*). Dengan begitu, adaptasi program aplikasi dari satu format resolusi layar ke format resolusi yang lain menjadi mudah.

Membuat UI di XML sangat mirip dengan membuat dokumen HTML, contoh sederhana :

```
<html>

  <head>

    <title>Page Title</title>

  </head>

  <body>

    The content of body element

  </body>

</html>
```

1) Hirarki Elemen Layar

Unit dasar dari aplikasi Android adalah *activity*, yaitu sebuah obyek dari kelas `Android.app.Activity`. *Activity* bisa melakukan banyak hal, tetapi dia sendiri tidak akan terlihat secara langsung di layar. Untuk membuat *activity* terlihat di layar adalah

dengan membuatnya sebuah *User Interface*, dalam hal ini dapat menggunakan *Views* dan *ViewGroups*.

a) *Views*

Views adalah obyek dari kelas (*base class*) `Android.view.Views`. Merupakan data struktur yang menyimpan layout dan konten untuk area layar. Obyek *View* menangani ukuran, *layout*, *drawing*, perpindahan fokus, *scrolling*, *key/gesture* untuk area layar.

Kelas *View* tersedia sebagai kelas induk (*base class*) untuk semua *widget*, implementasi dari subkelas yang menggambar elemen interaktif pada layar. *Widget* menangani sendiri bagaimana dia diimplementasikan, sehingga kita bisa memakainya untuk membuat UI secara lebih cepat. *List* dari *widgets* yang tersedia meliputi *TextView*, *EditText*, *Button*, *RadioButton*, *CheckBox*, *ScrollView*, dan lain-lain.

b) *View Groups*

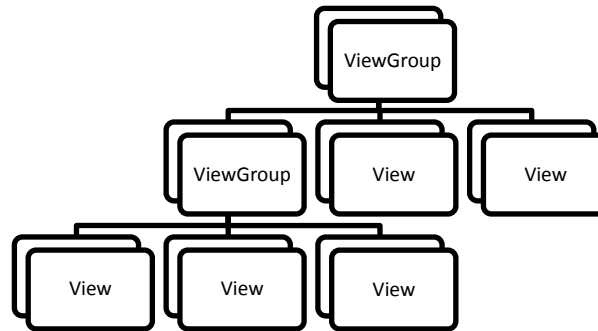
ViewGroups adalah obyek dari kelas `Android.view.ViewGroup`, dan dari namanya mengindikasikan *ViewGroup* adalah tipe khusus dari obyek *View* yang berfungsi untuk mewadahi dan

mengelola *subordinat Views* dan *ViewGroup* yang lain. *ViewGroup* membolehkan programmer untuk menambahkan struktur pada *User Interface* dan membuat elemen layar yang kompleks dan diberikan alamat sebagai satu kesatuan entitas.

Kelas *ViewGroups* tersedia sebagai kelas induk (*base class*) untuk *layout*, implementasi dari subkelas yang menyediakan tipe dasar dari layout layar. Dan layout memberikan cara bagaimana membangun struktur untuk *view*.

2) Hirarki *View* Android *User Interface*

Pada platform Android, programmer dapat mendefinisikan UI *Activity* dengan menggunakan struktur pohon *view (tree of view)* dan jaringan *ViewGroup* seperti yang terlihat pada gambar. Strukturnya bisa dibuat sederhana atau kompleks, sesuai dengan kebutuhan.



Gambar II.2. Hirarki View Android User Interface
 Sumber : Nicolash Gramlich, andbook! Android
 Programming release 002. <http://andbook.anddev.org/>, 2010.

Agar layar bisa menampilkan tampilan yang sesuai dengan hirarki *view*, *Activity* yang dibuat harus memanggil *method setContentView ()* dan *mempassing* referensi pada obyek yang ingin ditampilkan. Sistem Android kemudian akan menerima referensi itu dan menggunakannya sebagai nilai-nilai tampilan yang akan ditampilkan.

3) Android Manifest

File AndroidManifest.xml diperlukan oleh setiap aplikasi Android. Diletakkan pada folder *root* aplikasi, dan mendeskripsikan nilai global paket aplikasi kita. Termasuk *komponen* yang ada pada aplikasi (*activity*, *service*, dll) yang memang diekspos untuk sistem diluar aplikasi yang membutuhkan. Termasuk juga data pada tiap *Activity*, bagaimana menanganinya dan bagaimana *Activity* tersebut dijalankan (*launched*).

Hal yang penting yang harus diperhatikan dari *file* *AndroidManifest.xml* adalah *IntentFilters*. *IntentFilters* menjelaskan di mana dan kapan *Activity* dapat dijalankan. Ketika *Activity* atau sistem operasi akan melakukan sesuatu, misalnya membuka halaman web atau membuka layar kontak, itu akan membuat obyek *intent*.

Obyek *intent* ini akan menangani beberapa informasi yang mendeskripsikan apa yang akan dilakukan oleh aplikasi dan data apa yang dibutuhkannya untuk itu, selain juga beberapa informasi lain. Android membandingkan informasi pada obyek *intent* dengan *intentfilter* yang dibuka (*exposed*) oleh setiap aplikasi dan mencari *Activity* yang tepat untuk menangani data atau aksi (*action*) yang dibutuhkan oleh Aplikasi (*Activity*) yang memanggilnya. Jika terdapat lebih dari satu aplikasi yang menangani *intent* tersebut, pengguna (*user*) akan diberi peringatan, melalui pesan atau notifikasi mana yang akan dipilih.

Selain mendeklarasikan *Activity* Aplikasi, *Content Providers*, *Service*, dan *Intent Receivers*, harus juga ditetapkan *permission* di dalam *AndroidManifest.xml*.

a) Hampir setiap *AndroidManifest.xml* akan memasukkan deklarasi

namespace (*xmlns:Android*=<http://schemas.Android.com/>)

apk/res/ Android) pada elemen pertama. Ini membuat beberapa atribut standar Android tersedia di dalam *file*, yang akan digunakan untuk mensuply kebanyakan data untuk element di dalam *file*.

- b) Hampir setiap manifest memasukkan satu tag <application>, yang berisi beberapa tag yang menjelaskan tentang *Applications*, *IntentReceivers*, dll yang ada pada aplikasi.
- c) Jika programmer ingin agar *Activity* bisa langsung dibuka oleh user, programmer harus membuatnya mendukung MAIN dan LAUNCHER.

Struktur secara detail <tags> dari *file* AndroidManifest adalah :

- a) <manifest>, merupakan titik *root* utama dari AndroidManifest.xml. Berisi atribut paket (*packge-attribute*) yang menjadi titik dari semua paket pada Activity. Path Activity lain secara relatif akan mengacu pada nilai ini.

```
<manifest
xmlns:Android=http://schemas.Android.com/apk/res/Android
    package="org.anddev.Android.smsstrecher">
```

- b) <uses-permission>, menjelaskan tentang *security permission* yang harus diberikan agar aplikasi *activity* bisa secara benar beroperasi, misal ketika mengirim sms atau menggunakan

daftar kontak telepon. Permissi ini diberikan kepada user ketika penginstalan aplikasi. Jumlah : 0⁺

```
<uses-permission
```

```
Android:name="Android.permission.RECEIVE_SMS" />
```

- c) `<permission>`, mendeklarasikan *security permission* yang bisa digunakan untuk membatasi aplikasi yang mana yang dapat diakses *komponennya* atau fiturnya di dalam aplikasi yang dirancang atau di paket aplikasi lain. Jumlah : 0⁺
- d) `<instrumentation>`, mendeklarasikan kode dari *komponen* instrumen yang tersedia untuk menguji fungsionalitas dari paket aplikasi atau paket aplikasi yang lain. Jumlah : 0⁺
- e) `<application>`, Elemen *root* yang berisi deklarasi dari *level* aplikasi *komponen* yang terdapat pada paket aplikasi. Elemen ini juga bisa berisi global dan/atau atribut default dari aplikasi, seperti label, ikon, tema, permissi yang dibutuhkan, dll. Jumlah 0 atau 1.

```
<application Android:icon=@drawable/icon">
```

- f) `<activity>`, merupakan hal mendasar pada sebuah aplikasi untuk berinteraksi dengan user. Setiap *Activity* harus memiliki sebuah `<activity>` pada *file* manifest, apakah itu untuk diekspos keluar atau diperuntukkan hanya untuk paket aplikasi

itu sendiri. Jika *activity* tidak memiliki tag yang sesuai di dalam manifest, kita tidak bisa menjalankannya. Jumlah : 0⁺

Secara bebas, untuk mendukung *late runtime lookup*, kita bisa menempatkan elemen `<intent-filter>` 1⁺ untuk mendeskripsikan aksi yang didukung *activity*.

g) `<intent-filter>`, mendeklarasikan *intent* seperti apa yang didukung oleh *komponen*. Sebagai tambahan kita bisa menempatkan nilai pada elemen ini, atribut bisa diberikan di sini untuk mensupply label, ikon, dan informasi lain untuk aksi (*action*) yang akan dilakukan.

h) `<action>`, tipe aksi (*action-type*) yang didukung oleh *komponen*. Contoh :

```
<action Android:name="Android.intent.action.MAIN">
```

i) `<category>`, kategori (*category-type*) yang didukung oleh *komponen*. Contoh :

```
<category
```

```
Android:name="Android.intent.category.LAUNCHER">
```

j) `<data>`, tipe MIME, skema URI, authority URI, atau path URI yang didukung *komponen*.

k) `<meta-data>`, menambahkan bagian baru metadata kepada *activity*, dimana client akan mendapatkan nilainya kembali.

```
ComponentInfo.metaData.
```

- l) `<receiver>`, sebuah *intentReceiver* mengizinkan aplikasi untuk diberikan informasi mengenai perubahan data atau aksi yang terjadi, bahkan jika sedang berjalan (*running*). Seperti juga dengan tag `<activity>`, secara kita dapat menempatkan lebih dari satu elemen `<intent-filter>` yang didukung oleh receiver atau memasukkan nilai pada `<meta-data>`.

```
<receiver Android:name=".SMSReceiver">
```

- m) `<service>`, merupakan *komponen* yang dapat berjalan pada *background*. Seperti tag *activity*, secara opsional kita dapat menempatkan satu atau lebih elemen `<intent-filter>` yang didukung oleh *service* atau nilai `<meta-data>`.
- n) `<provider>`, merupakan *komponen* yang mengelola data (*persistent data*) dan mempublishnya untuk diakses oleh aplikasi lain. *Developer* secara opsional dapat meletakkan satu atau lebih nilai `<meta-data>`, seperti yang dijelaskan pada `activity <meta-data>`. (Gramlich, 2010 : 19-28).

4) *Resources* dan R.Java

a) *Resources*

Resources adalah *file* eksternal (*non-code files*) yang digunakan oleh *source code* aplikasi dan dikompile ke dalam aplikasi pada saat *build*. Android mendukung banyak macam *file resources*, termasuk *file* XML, PNG, dan JPEG.

Resource ditempatkan di luar *source code*, dan *file* XMP di kompile ke dalam binary, format yang bisa cepat di load untuk alasan efisiensi. String dikompres ke dalam format penyimpanan yang lebih efisien.

(1) Daftar *Resources*

Tipe *resource* dan tempat di mana mereka ditempatkan :

- a. Layout-*files* : “/res/layout/”
- b. Images : “/res/drawable/”
- c. Animation : “/res/anim/”
- d. Styles, string dan array : “/res/values/”

Nama *file* tidak mesti persis dengan :

- a. ‘array.xml’ untuk mendefinisikan array
- b. ‘colors.xml’ untuk mendefinisikan warna #RGB, #ARGB, #RRGGBB, #AARRGGBB
- c. ‘dimens.xml’ untuk mendefinisikan dimensi
- d. ‘strings.xml’ untuk mendefinisikan string

- e. 'styles.xml' untuk mendefinisikan obyek
 - f. Raw *file* seperti mp3 atau video : “/res/raw/”
- (2) Menggunakan *Resources* di dalam *source code*

Menggunakan *resource* di dalam *source code* adalah dengan mengetahui ID *resource* dan tipe dari obyek *resource* tersebut. Adapun syntax untuk mengacu kepada *resource* :

```
R.resource_type.resource_name atau
Android.R.resource_type.resource_name
```

Dimana *resource_type* adalah subkelas (*subclass*) dari *R* yang merupakan tipe spesifik dari *resource.resource_name* merupakan atribut dari *resource* yang didefinisikan pada *file XML*, atau *file name* (tanpa ekstensi) untuk *resource* yang didefinisikan oleh tipe *file* lain. Setiap tipe dari *resource* akan ditambahkan pada subkelas *R* yang spesifik, tergantung dari tipe *resources*.

Resources yang di *kompile* sendiri oleh aplikasi dapat direfer tanpa nama paket (seperti *R.resource_type.resource_name*).

Android memiliki beberapa standar *resource*, seperti *style* latar, dan *background button*. Untuk merefer pada kode itu, kita harus memastikan kualifikasinya dengan Android, seperti contoh :

```
Android.R.drawable.button_background.
```

(3) *Resource Reference*

Nilai yang dimasukkan pada atribut dapat juga direfer pada *resource* yang lain. Seringkali ini dilakukan pada *file layout* untuk memasukkan *string* sehingga bisa *dilocalized* dan *image* yang berada pada *file* lain, walaupun juga bisa dilakukan pada tipe *resource* lain misalnya warna dan nilai integer.

Sebagai contoh, jika kita memiliki *resource* warna, kita bisa membuat *file layout* yang mengeset ukuran dari *text color* menjadi nilai yang dimiliki oleh *resource* itu.

```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@color/opaque_red"
    Android:text="Hello, World!" />
```

Seperti yang terlihat prefik “@” digunakan untuk memperkenalkan *resource* reference, teks yang menyertai kemudian adalah nama dari *resource* di dalam format @[package:]type/name. Pada kasus ini paket tidak dispesifikasikan karena *resource* direferensi di dalam paket itu sendiri. Untuk mereference sistem *resource*, dapat dilakukan dengan cara :

```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@Android:color/red"
    Android:text="Hello_world!" />
```

atau dalam contoh lain, sebaiknya selalu menggunakan *resource* reference ketika mensupply string ke dalam *file* layout sehingga *resource* tersebut bisa di lokalisir (*localized*)

```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@Android:color/red"
    Android:text="@string/Hello_world!" />
```

b) R.Java

R.Java merupakan *file* yang digenerate secara otomatis, yang mengindeks semua *resource* dari proyek aplikasi. Kelas ini bisa digunakan ke dalam *source code* sebagai cara pintas untuk merefer ke *resource* yang di masukkan ke dalam proyek aplikasi. Ini sangat bermanfaat apabila dikaitkan dengan fitur IDE *code-completion* seperti Eclipse karena ini membuat *developer* secara cepat dan interaktif menemukan spesifik *resource*. Selain itu, *developer* bisa meningkatkan waktu kompilasi karena *resource* yang akan dipakai benar-benar ada.

10. Unified Modeling Language (UML)

Unified Modeling Language atau lebih sering disebut UML adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang

membutuhkan pemodelan. Bagian-bagian utama dari UML adalah *view*, *diagram*, model *element*, dan *general mekanisme*.

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain: *use case view*, *logical view*, *component view*, *concurrency view* dan *deployment view*.

b. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.

View ini digambarkan dalam *use case diagram* dan kadang-kadang dengan *activity diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*) dan penguji sistem (*tester*).

c. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object* dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *objek* mengirim pesan ke *objek* lain dalam suatu fungsi tertentu.

View ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state*, *sequence*, *collaboration*, dan *activity diagram* untuk model

dinamisnya. View ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

d. *Component view*

Mendeskripsikan implementasi dan ketergantungan modul. *Komponen* yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya *komponen* dan informasi administratif lainnya. View ini digambarkan dalam component view dan digunakan untuk pengembang (*developer*).

e. *Concurrency view*

Membagi sistem ke dalam proses dan prosesor. View ini digambarkan dalam diagram dinamis (*state, sequence, collaboration, dan activity diagrams*) dan diagram implementasi (*component dan deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

f. *Deployment view*

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. View ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).


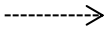
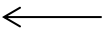
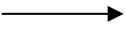
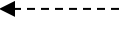
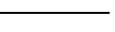
g. Diagram

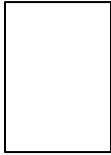
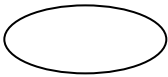


Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

1. Use case Diagram

Use case adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*. Berikut ini dapat dilihat simbol-simbol dari *use case* itu sendiri. Sebuah simbol dapat memperjelas diagram alur dari sebuah sistem yang dirancang.

Tabel II.1. Simbol *Use case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana <i>objek</i> anak (<i>descendent</i>) berbagi perilaku dan struktur data dari <i>objek</i> yang ada di atasnya <i>objek</i> induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara <i>objek</i> satu dengan <i>objek</i> lainnya.

7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan sumber daya komputasi

2. Class Diagram

Class diagram adalah deskripsi kelompok *objek-objek* dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class* diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class*

diagram. *Class* diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

3. **Component Diagram**

Component software merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. *Komponen* merupakan implementasi *software* dari sebuah atau lebih *class*. *Komponen* dapat berupa *source code*, *komponen* biner, atau *executable component*. Sebuah *komponen* berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *komponen view*. Sehingga *komponen* diagram merepresentasikan dunia riil yaitu *komponen software* yang mengandung *komponen*, *interface* dan *relationship*.

4. **Deployment Diagram**

Menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *objek* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan *komponen*.

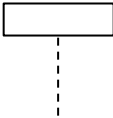

5. State Diagram


Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa object lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

6. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *objek* juga interaksi antara *objek*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Pada halaman disebelah dapat dilihat simbol-simbol dari *sequence* diagram :

Tabel II.2. Simbol *Sequence* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	<i>Objek entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi-informasi tentang aktifitas yang terjadi

3		<i>Objek</i>	Spesifikasi dari field-field atau sub menu yang dapat diakses oleh <i>actor</i>
---	---	--------------	---

7. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*.


Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *objek* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.





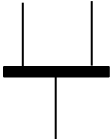
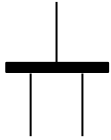
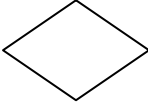
8. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Berikut adalah simbol-simbol yang sering digunakan pada tahap perancangan *activity diagram* :

Tabel II.3. Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.

2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Titik awal
4		<i>Activity Final Node</i>	Titik akhir
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Join</i>	Penggabungan beberapa <i>state</i>
7		<i>Rake</i>	Percabangan beberapa <i>state</i>
8		<i>Decision</i>	Digunakan untuk penyeleksian kondisi didalam <i>activity diagram</i>

BAB III

METODE PENELITIAN

A. Lokasi Penelitian

Adapun tempat yang digunakan untuk melakukan penelitian adalah beberapa rumah makan yang ada di daerah Makassar yaitu resep masakan khas Makassar, serta beberapa resep masakan diambil dari literatur yang ada.

B. Jenis Penelitian

Jenis penelitian yang digunakan dalam skripsi ini adalah jenis penelitian deskriptif kualitatif. Penelitian deskriptif kualitatif adalah penelitian yang menggambarkan fakta, keadaan, variabel dan fenomena-fenomena *objek* ketika penelitian berlangsung sesuai dengan apa adanya. Dalam hal ini lebih memprioritaskan pada menu kuliner yang lebih *bervariasi* sehingga masyarakat dapat mengetahui secara pasti bahan-bahan dan cara memasak dari setiap resep makanan yang ada pada aplikasi.

C. Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Penelitian Kepustakaan (*Library Research*)

Yaitu penelitian yang dilakukan dengan cara mengumpulkan data yang terdapat dalam buku-buku *literatur*, peraturan perundang-undangan, majalah,

surat kabar, hasil seminar dan sumber lain yang terkait dengan perancangan aplikasi menu kuliner.

2. Penelitian Lapangan (*Field Research*)

Yaitu melakukan penelitian langsung ke lokasi penelitian untuk memperoleh data yang valid mengenai *objek* yang diteliti. Dalam hal ini tentang kuliner yang ada di daerah Makassar dan sekitarnya. Tidak menutup kemungkinan akan ada masakan dari berbagai daerah lain yang akan dimasukkan dalam basis data kuliner.

D. Hardware dan Software

1. *Hardware* yang digunakan dalam penelitian ini adalah :

a. *Handset Android*

Dalam penelitian ini handset yang digunakan adalah Samsung Tab 2 GT-P3100 Processor Qualcomm MSM7227 ARM11 800 MHz, RAM 1 GB dengan Sistem Operasi Android versi 4.0 (Ice Cream).

b. Laptop/Komputer

Dalam penelitian ini laptop yang digunakan laptop Anote M550SE dengan Processor Intel T2130 (1.86 Ghz, RAM 2 GB). Dengan Sistem Operasi Microsoft Windows Seven, 32 bit.

2. *Software* yang digunakan dalam penelitian ini adalah :

a. Eclipse dan Android SDK

Dalam penelitian ini, menggunakan aplikasi Eclipse Juno dan Android SDK dalam membangun aplikasi.

b. JDK

Dalam penelitian ini, *software* development kit yang digunakan dalam pemrograman bahasa java. JDK yang digunakan ada versi 722-windows i586.

E. System Requirement

Aplikasi yang dirancang adalah aplikasi *stand alone* dan *offline*, sedangkan kebutuhan sistem minimal dari aplikasi ini adalah :

1. Perangkat *mobile*

Sistem Operasi Android versi 2.2 (Froyo).

F. Metode Pengujian Perangkat Lunak

Pengujian dimaksudkan untuk mengetahui apakah perangkat lunak yang dibuat telah memenuhi tujuan dan menunjukkan bahwa aplikasi yang dirancang memiliki jaminan kualitas yang baik. Secara umum teknik pengujian dilakukan dengan teknik *white box* dan *black box*. Dalam penelitian ini metode yang digunakan adalah metode pengujian *black box*, dimana teknik ini merupakan pengujian analisa

sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program. ”*Black box* merupakan salah satu metode pengujian yang memecah atau membagi domain input dari program ke dalam kelas-kelas data sehingga *test case* dapat diperoleh (Pressman, 2002 : 532)

Pada *Black-Box Testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan (Al-Fatta, 2007 : 172).

BAB IV

ANALISIS DAN DESAIN SISTEM

A. Analisis Sistem

Menjelaskan langkah dan tujuan analisis terhadap objek penelitian yang akan diteliti.

1. Analisis Kebutuhan Sistem

a. Kebutuhan Perangkat Keras

Perangkat *mobile* : *Smartphone* yang memiliki sistem operasi Android versi 2.2 (Froyo) sampai versi 4.2 (Jellybean).

b. Kebutuhan Perangkat Lunak

Software JRE (*Java Runtime Environment*) dan Java SDK (*Standart Development Kit*), JRE digunakan untuk menjalankan aplikasi yang berbasis java, pembangunan sistem berupa aplikasi yang nantinya akan digunakan untuk berhubungan dengan internet.

Eclipse Juno dan Android SDK adalah aplikasi yang dibutuhkan dalam pembuatan aplikasi berbasis Android. Eclipse merupakan IDE untuk pembuatan aplikasi dan Android SDK merupakan library yang harus dimiliki untuk menjalankan program.

c. Kebutuhan Informasi

Informasi yang akan dihasilkan dari aplikasi ini dapat mempermudah masyarakat untuk mengakses informasi kuliner khususnya cara pembuatan kuliner beserta bahan-bahannya melalui *handset* Android.

d. Kebutuhan Pengguna

Dalam penelitian ini yang berperan dalam pembangunan aplikasi adalah *masyarakat*. Hal – hal yang menjadi aspek kebutuhan *masyarakat* yaitu :

- i. Kebutuhan *masyarakat* ialah aspek kemudahan dalam mengakses informasi-informasi kuliner berupa bahan-bahan dan cara membuat masakan dimanapun dan kapanpun.

2. Analisis Kelayakan Sistem

a. Kelayakan Teknologi

Teknologi yang akan digunakan pada aplikasi berbasis *mobile* untuk mengakses informasi tentang menu kuliner yaitu teknologi berbasis pemrograman *mobile*. Teknologi ini sudah umum digunakan dalam berbagai bidang dan telah dikembangkan ke sistem yang lebih kompleks. Dewasa ini, Teknologi berbasis pemrograman *mobile* ini mudah ditemukan dalam kehidupan sehari-hari. Karena perangkat *mobile* seperti *handphone* dan *smartphone* adalah perangkat yang paling dekat dengan manusia.

b. Kelayakan Hukum

Pembuatan aplikasi berbasis android untuk mengakses informasi tentang menu kuliner serta cara membuatnya tidak akan melanggar

hukum atau peraturan – peraturan yang berlaku baik yang telah ditetapkan oleh pemerintah.

c. Kelayakan Operasional

Pada kelayakan Operasional dibagi menjadi 2 aspek yaitu:

1. Aspek teknis

Dengan dibangunnya aplikasi yang mengkombinasikan antara *menu kuliner* dengan perkembangan teknologi, *user* (masyarakat) akan diberi kemudahan dalam mengakses informasi menu kuliner khususnya tentang langkah-langkah dan bahan yang digunakan dalam membuat masakan.

2. Aspek psikologis

Sistem ini mudah diterapkan, karena tidak membutuhkan pelatihan yang rumit.

B. Deskripsi Umum Perangkat Lunak

Desain adalah langkah pertama dalam fase pengembangan bagi setiap produk atau sistem yang direkayasa. Desain dapat didefinisikan sebagai “proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya” .

Tujuan desainer adalah untuk menghasilkan suatu model atau representasi dari entitas yang kemudian akan dibangun. Proses dimana model dikembangkan menggabungkan intuisi dan penilaian berdasarkan pengalaman dalam membangun entitas yang sama, serangkaian prinsip dan heuristik yang menuntun cara dimana model disusun, serangkaian kriteria yang memungkinkan penilaian terhadap kualitas, dan suatu proses iterasi yang membawa kepada suatu representasi desain akhir (Pressman, 2002 : 399).

Output utama dari tahapan desain perangkat lunak adalah spesifikasi desain. Spesifikasi ini meliputi spesifikasi desain umum yang akan disampaikan kepada stakeholder sistem dan spesifikasi desain rinci yang akan digunakan pada tahap implementasi. Spesifikasi desain umum hanya berisi gambaran umum agar *stackholder* sistem mengerti akan seperti apa perangkat lunak yang akan dibangun. Spesifikasi desain rinci atau kadang disebut desain arsitektur rinci perangkat lunak diperlukan untuk merancang sistem sehingga memiliki konstruksi yang baik, proses pengolahan data yang tepat dan akurat, bernilai, memiliki aspek *user friendly* dan memiliki dasar-dasar untuk pengembangan selanjutnya.

Desain perangkat lunak berada pada inti teknik dari proses rekayasa perangkat lunak dan diaplikasikan tanpa memperhatikan model proses perangkat lunak yang digunakan. Begitu persyaratan perangkat lunak telah mulai dianalisis dan ditentukan, maka desain perangkat lunak menjadi yang pertama dari tiga aktifitas teknik (desain, pembuatan kode, dan pengujian) yang diperlukan untuk membangun dan menguji

perangkat lunak. Masing-masing aktifitas memindahkan informasi dengan suatu cara yang terutama menghasilkan perangkat lunak komputer yang tervalidasi.

Aplikasi teknik pencarian cerdas menu kuliner yang dirancang untuk penyelesaian tugas akhir ini diberi nama Libar Cook. Pemberian nama Libar Cook ini bermula dari keinginan penulis untuk membuat kreasi masakan yang akan disajikan untuk sang suami, karena takut bahan pangan yang dibutuhkan dalam membuat masakan baru akan sia-sia jika masakan yang dibuat ternyata gagal. Maka penulis berinisiatif membuat sebuah aplikasi menu kuliner. Libar itu sendiri merupakan gabungan kata dari *zodiak* penulis dengan *zodiak* sang suami. “Lib” untuk *zodiak* Libra, dan “ar” untuk *zodiak* Aries. Kemudian digabungkan menjadi “Libar”. Cook sendiri diadopsi dari bahasa Inggris yang berarti memasak. Untuk selanjutnya aplikasi teknik pencarian cerdas menu kuliner disebut dengan Libar Cook.

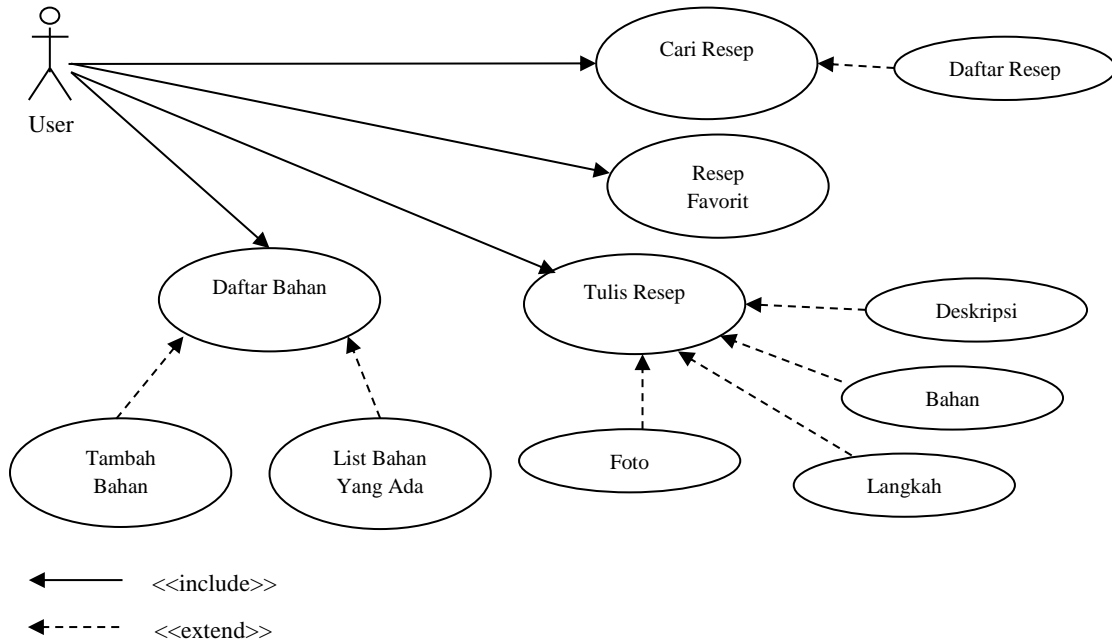
C. Perancangan Sistem

1. Proses

Tahap perancangan dimulai dengan pembuatan *UML* atau *Unified Modelling Language* untuk mengetahui perilaku bagaimana sistem bereaksi. Perilaku ini merupakan aktifitas sistem yang bisa dilihat dari luar dan bisa diuji.

Untuk memperjelas *UML* dari aplikasi Libar Cook sendiri, dapat dilihat pada *Use Case*, *Class Diagram*, *Activity Diagram* serta *Sequence Diagramnya*, berikut adalah gambar *Use Case* dari aplikasi yang dirancang.

a. Use Case diagram aplikasi Libar Cook yang dirancang

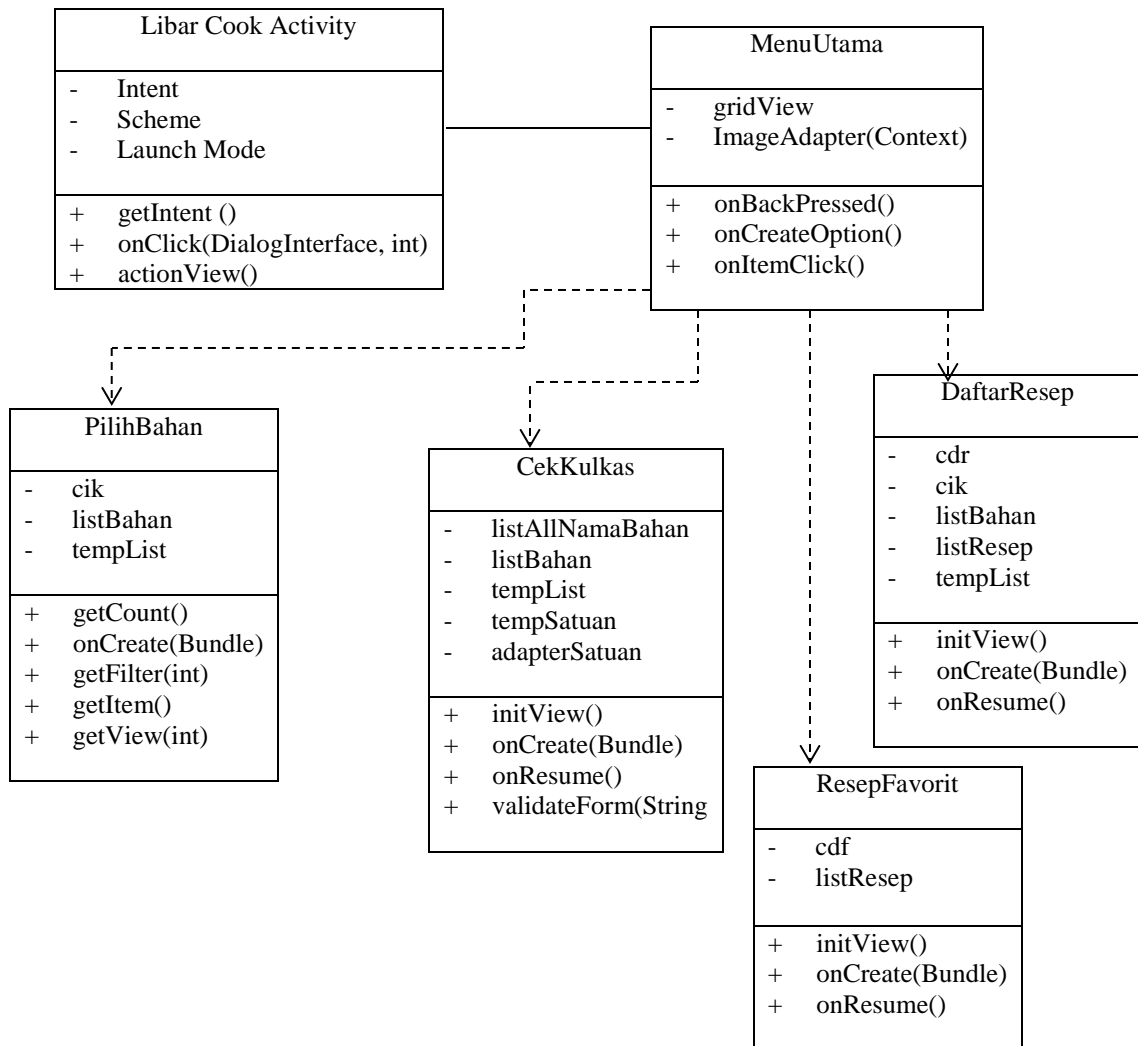


Gambar IV.1. Use Case Diagram Libar Cook

Ketika *user* mengakses aplikasi Libar Cook, maka *user* dapat mengakses atau memilih *fitur-fitur* yang ada pada aplikasi. Sebagai contoh, *user* memilih *fitur* tulis resep. Maka secara otomatis sistem akan menampilkan sub bagian dari tulis resep. Mulai dari deskripsi masakan, bahan, cara membuat masakan serta sampel foto yang dapat di *upload* kedalam sistem.

b. Class Diagram

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Dalam *class diagram* akan ditampilkan beberapa *class* yang mewakili aplikasi secara global.



Gambar IV.2. Class Diagram Libar Cook

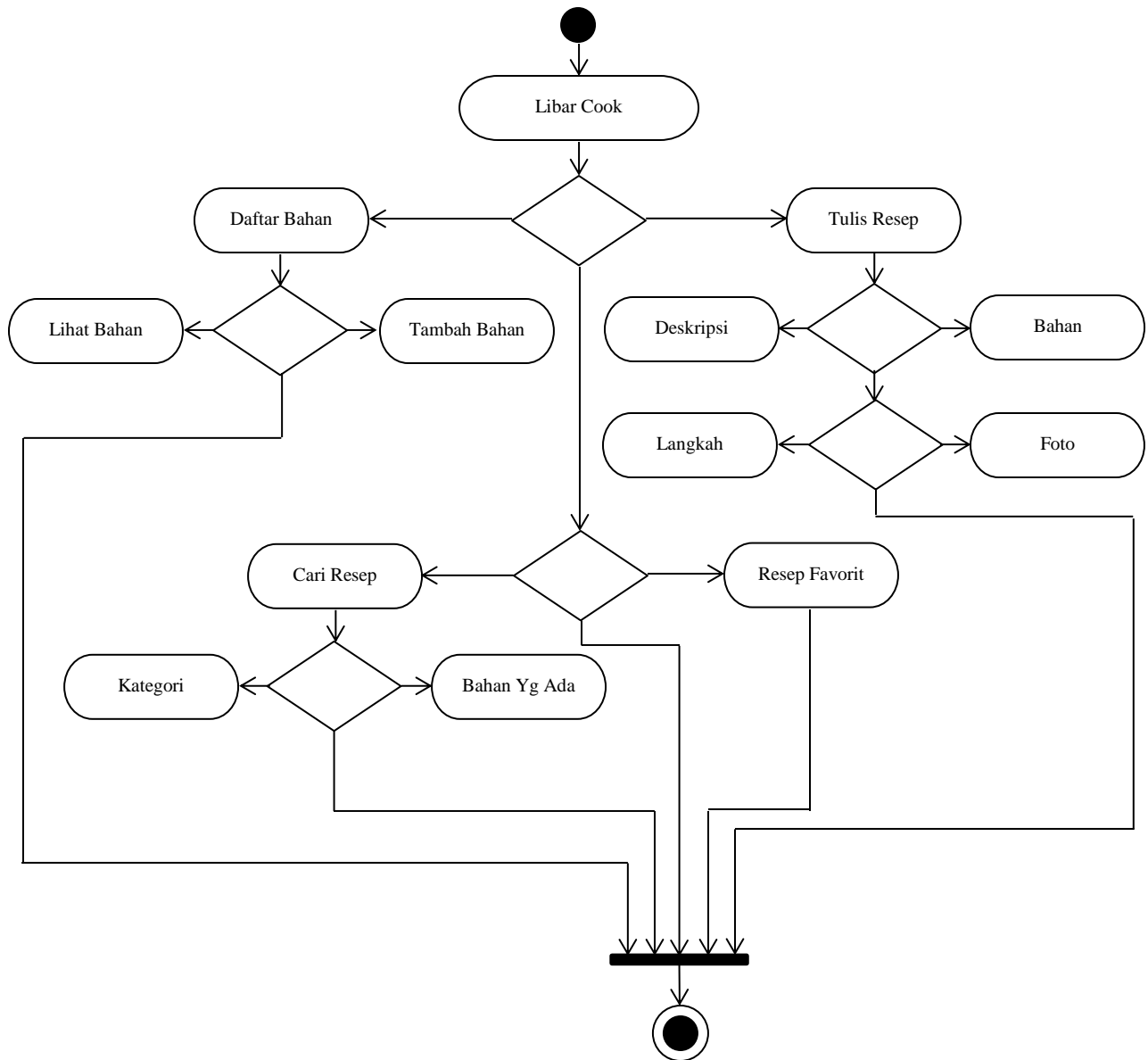
Gambar *class diagram* diatas merupakan alur utama dari *class-class* yang ada pada aplikasi ini. *Class MenuUtama* merupakan *class* global yang membawahi fungsi utama untuk merelasikan fungsi dari *class-class* bawaan seperti *class PilihBahan*, *class CekKulkas*, *class ResepFavorit* dan *class DaftarResep*.

c. Activity Diagram

Activity diagram merupakan bentuk *flow diagram* yang memodelkan alur kerja sebuah proses dan urutan aktivitas sebuah proses. Diagram ini mirip dengan *flowchart* karena dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya. *Activity diagram* juga berguna untuk menggambarkan perilaku paralel atau bagaimana perilaku dalam berbagai *use case* berinteraksi.

Activity diagram pencarian cerdas menu kuliner berikut menjelaskan alur sistem, yang dimulai dengan menampilkan menu utama yang terdiri dari beberapa pilihan, yaitu daftar bahan, pencarian resep, membuat resep sendiri hingga membuat beberapa resep yang ada menjadi resep favorit. Untuk *item* daftar bahan, sistem akan menampilkan bahan-bahan yang tersedia. Untuk bahan yang belum terdapat pada daftar bahan, *user* dapat menambahkannya sendiri dengan memilih tombol tambah bahan.

Berikut adalah *activity diagram* secara keseluruhan proses yang terjadi di dalam sistem aplikasi teknik pencarian cerdas menu kuliner meliputi *user* (masyarakat).



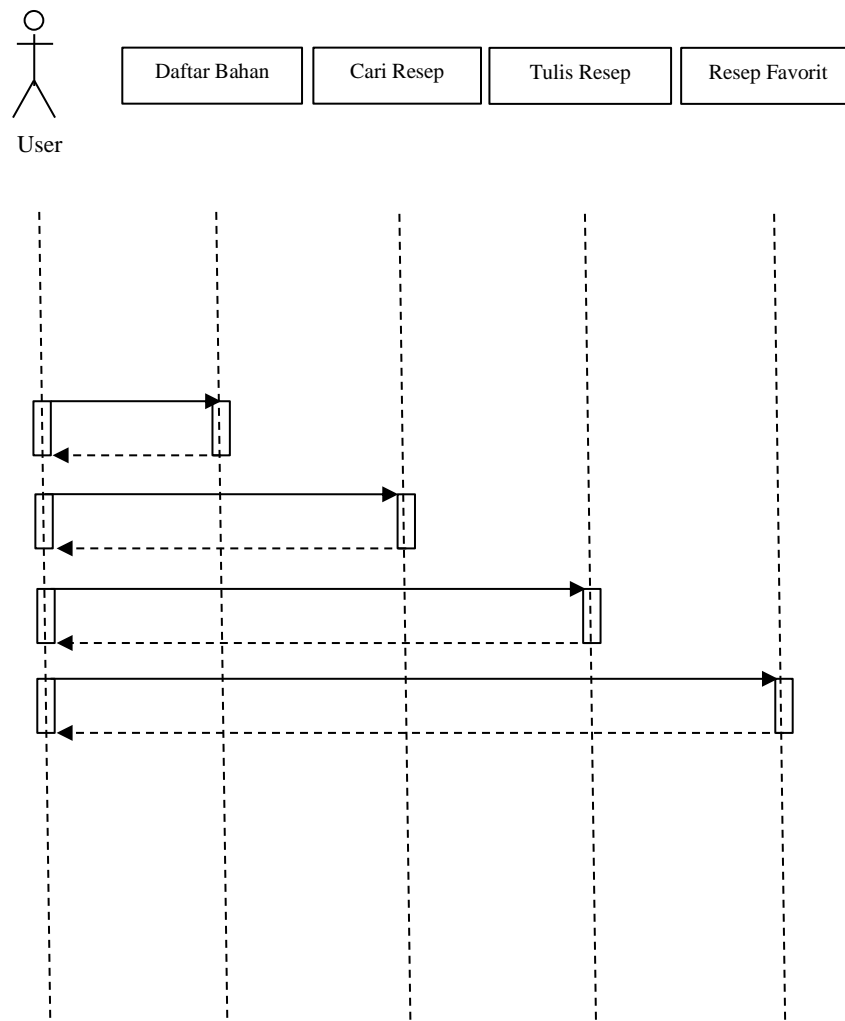
Gambar IV.3. Activity Diagram Libar Cook

Dari *activity diagram* diatas, dapat diketahui bahwa ketika aplikasi ini dijalankan, sistem akan memberikan pilihan, yaitu daftar bahan, cari resep, tulis resep dan resep favorit. Dari empat pilihan yang ada, yang menjadi kunci utama aplikasi ini

adalah daftar bahan. Karena semua bahan pangan yang dibutuhkan untuk membuat sebuah masakan terdapat pada daftar bahan dari aplikasi ini.

d. Sequence Diagram

Sequence diagram menjelaskan interaksi obyek yang tersusun dalam suatu urutan waktu. Dalam *sequence diagram* diperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*.

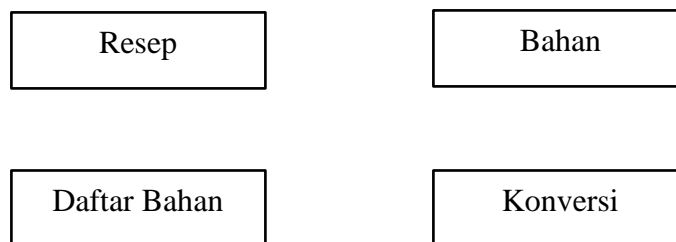


Gambar IV.4. Sequence Diagram

Dari gambar *sequence diagram* aplikasi ini, dapat diketahui bahwa ketika *user* telah menjalankan aplikasi Libar Cook, maka semua *fitur* pelengkap dari aplikasi ini dapat diakses seperti sebuah satu kesatuan yang saling berinteraksi. Ketika *user* memilih daftar bahan, maka sistem akan melakukan *request* terhadap *database* bahan pangan, kemudian merespon permintaan tersebut sehingga menampilkan semua bahan pangan yang terdapat pada daftar bahan pada aplikasi ini.

Untuk penelusuran lebih lanjut terhadap *database* dari aplikasi ini, maka dibuat *Entity Relationship Diagram (ERD)*. Berikut adalah rancangan *ERD* dari aplikasi Libar Cook yang dirancang :

Tahap 1: Penentuan Entitas



Gambar IV.5. Entitas ERD

Keterangan :

Resep:

- Id_Resep: identitas resep (PK)
- Nama: nama resep
- Deskripsi: deskripsi resep
- Langkah: langkah membuat resep masakan
- Kategori: kategori resep
- Flagfavorit: tanda favorit resep
- Foto: gambar resep

Bahan:

- Id_Bahan: identitas bahan (PK)
- Koderesep: kode nama resep
- Nama: nama bahan
- Jumlah: jumlah bahan
- Satuan: satuan bahan

Daftar Bahan:

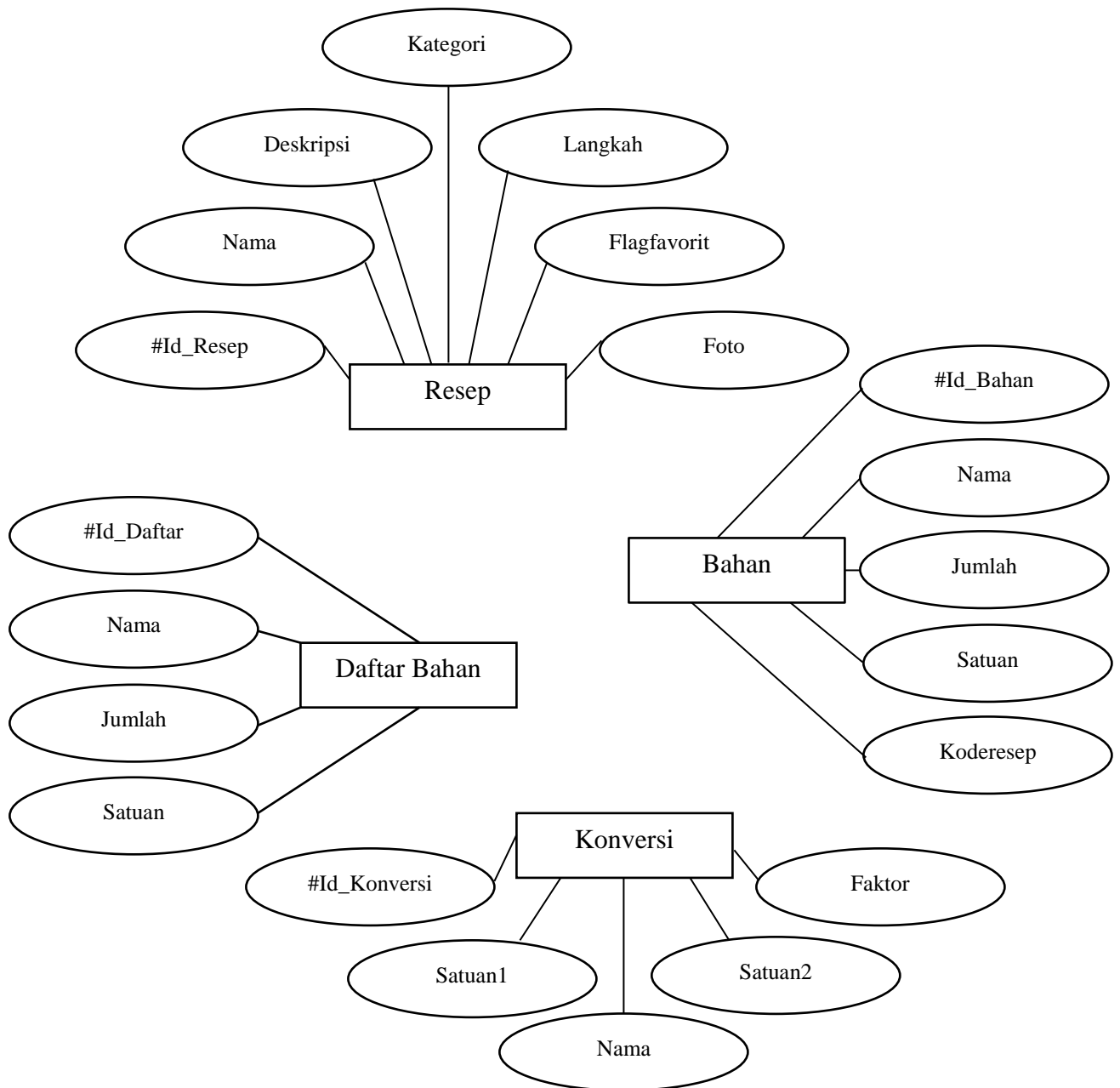
- Id Daftar: identitas daftar bahan (PK)
- Nama: nama daftar bahan
- Jumlah: jumlah daftar bahan
- Satuan: satuan daftar bahan

Konversi:

- Id_Konversi: identitas konversi (PK)
- Nama: nama konversi
- Satuan1: satuan pertama konversi
- Satuan2: satuan kedua konversi
- Faktor: faktorisasi konversi

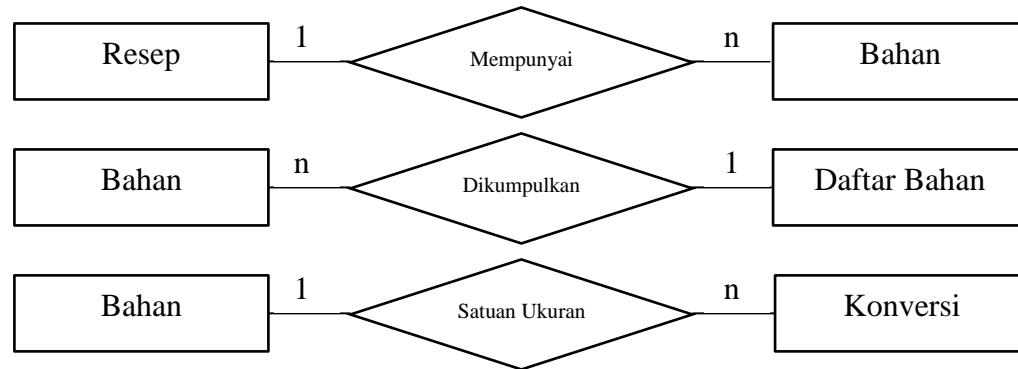
Untuk tahap awal penentuan *entitas* yang akan digunakan dan dijabarkan dalam tahap-tahap berikutnya. Adapun *entitas* yang digunakan adalah *entitas* resep, *entitas* bahan, *entitas* daftar bahan dan *entitas* konversi. Yang mana setiap *entitas* memiliki keterkaitan dengan *entitas* yang lainnya. Sebagai contoh *entitas* resep berkaitan dengan *entitas* bahan, sedangkan *entitas* bahan terkait langsung dengan *entitas* daftar bahan dan *entitas* konversi

Tahap 2: Penentuan Atribut



Gambar IV.6. Atribut ERD

Tahap 3: Penentuan Kardinalitas Relasi



Gambar IV.7. Kardinalitas Relasi ERD

Hubungan :

A. resep terdiri dari bahan

- Tabel utama: **resep**
- Tabel kedua: **bahan**
- Relationship: One to many (1:n)
- Atribut penghubung: **id_resep** (FK **id_resep** di **bahan**)

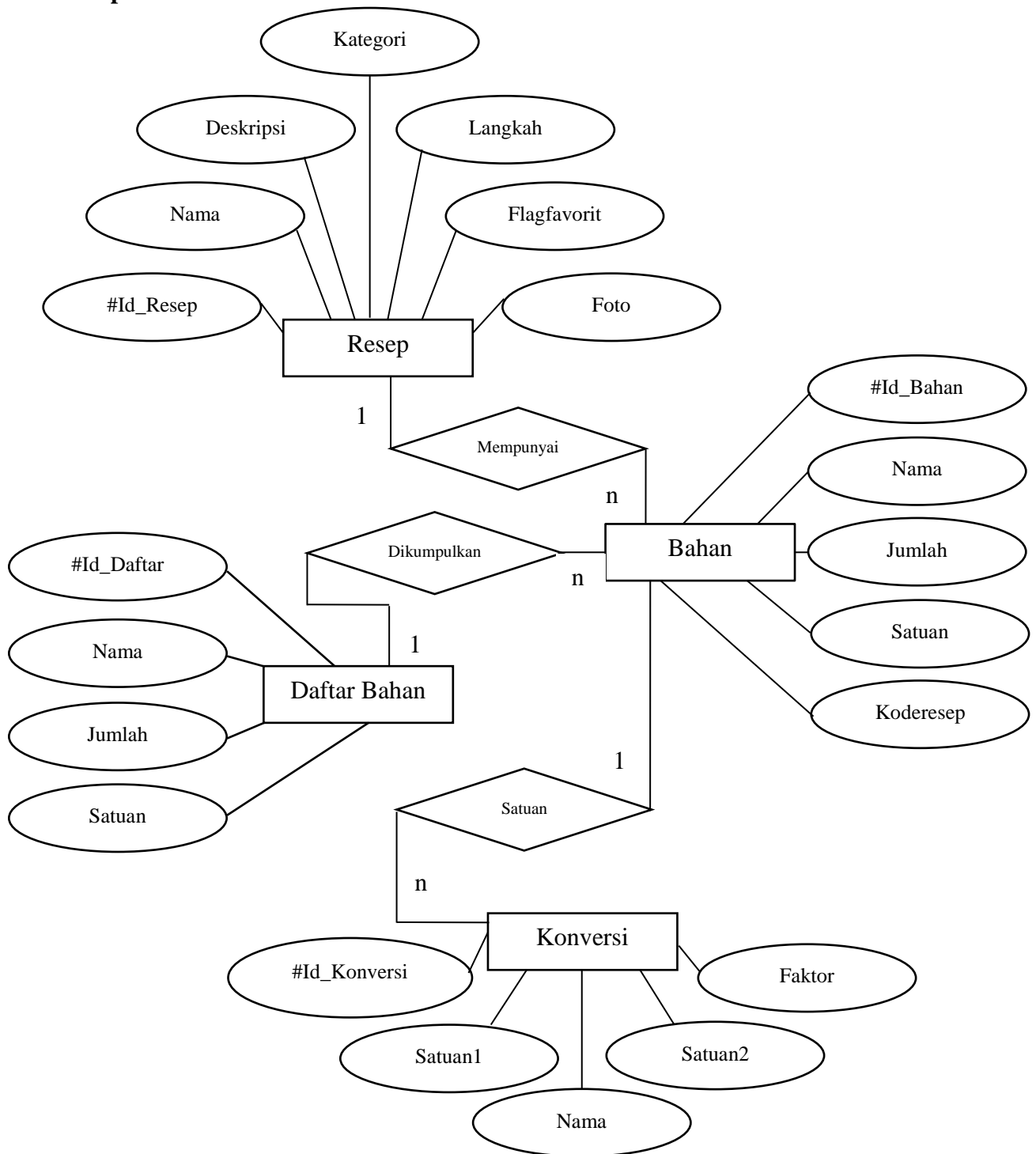
B. bahan disatukan daftar bahan

- Tabel utama: **bahan**
- Tabel kedua: **daftar bahan**
- Relationship: One to one (1:1)
- Atribut penghubung: **id_bahan** (FK **id_bahan** di **daftar bahan**)

C. bahan satuan konversi

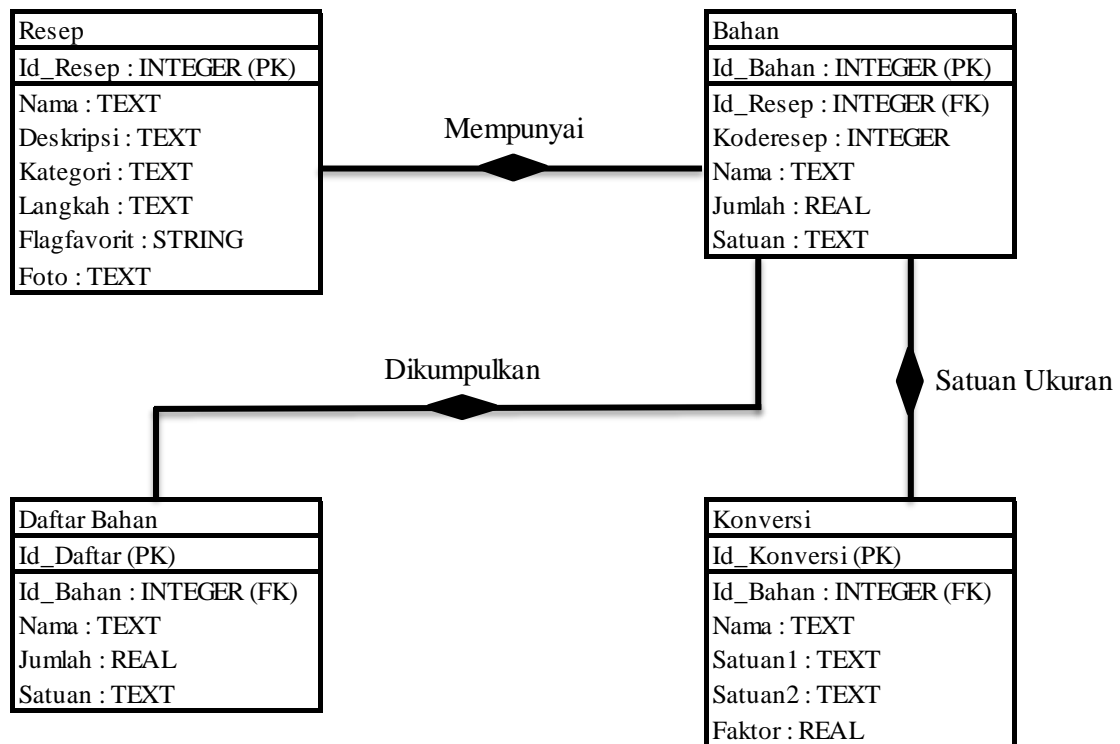
- Tabel utama: **bahan**
- Tabel kedua: **konversi**
- Relationship: One to many (1:n)
- Atribut penghubung: **id_bahan** (FK **id_bahan** di **konversi**)

Tahap 4: Pembuatan ERD



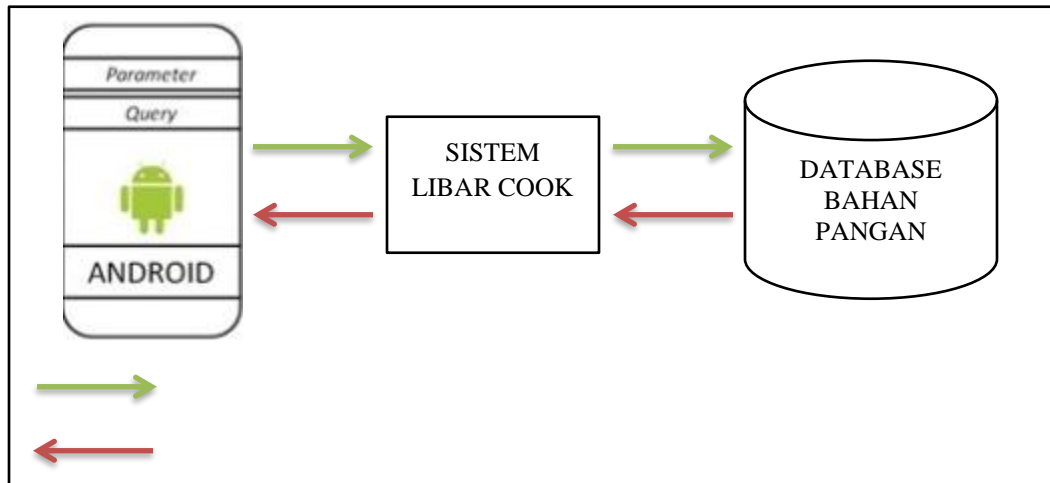
Gambar IV.8. ERD Libar Cook

Setelah pembuatan ERD selesai, maka selanjutnya adalah tahap pembuatan *EER (Enhanced Entity Relationship) diagram*. EER dibuat untuk lebih memperjelas relasi antara entitas yang ada dengan lebih tertuju pada *Primary Key* dan *Foreign Key* dari masing-masing *entitas*.



Gambar IV.9. EER Libar Cook

Aplikasi pencarian cerdas menu kuliner yang telah dibuat diakses pada kondisi *offline*. Berikut adalah diagram rancangan sistem yang dirancang.



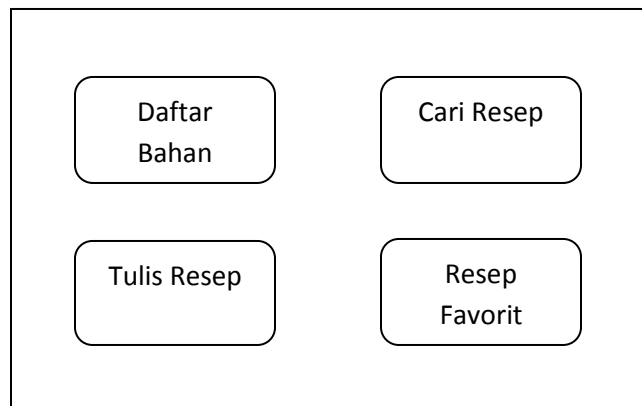
Gambar IV.10. Diagram Rancangan Sistem

Dapat dilihat pada gambar di atas, aplikasi menu kuliner ini digunakan dengan *offline* tanpa harus menggunakan koneksi *internet*. Kondisi ini memungkinkan *user* menggunakan fitur-fitur yang ada secara lengkap. Sistem memberikan *request* terhadap database bahan pangan ketika dibutuhkan, yakni ketika *user* mengakses *fitur* daftar bahan dan *fitur* cari resep.

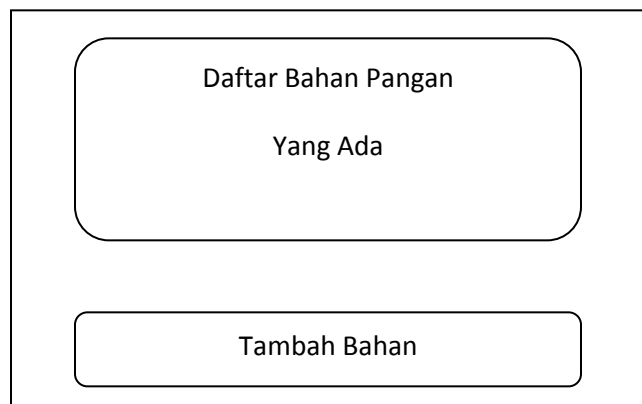
2. Perancangan Antarmuka / *Interface*

Perancangan antarmuka mengandung penjelasan tentang rancangan antar muka dari aplikasi perancangan form-form yang ada dalam aplikasi.

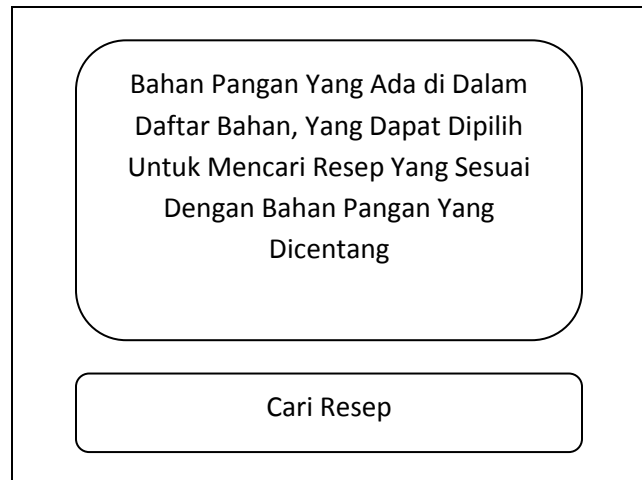
Adapun rancangan *interface* pada aplikasi Libar Cook ini adalah :



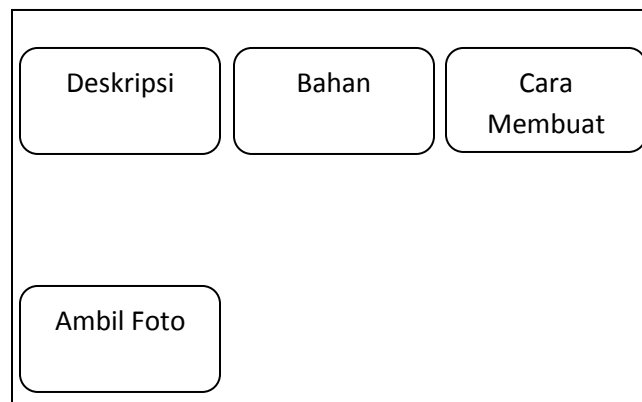
Gambar IV.11. Rancangan Jendela Utama



Gambar IV.12. Rancangan Jendela Daftar Bahan



Gambar IV.13. Rancangan Jendela Cari Resep



Gambar IV.14. Rancangan Jendela Tulis Resep

Untuk rancangan antarmuka / *interface* dari aplikasi pencarian cerdas menu kuliner ini digunakan bahasa pemrograman Java karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa *platform* sistem operasi yang berbeda, seperti Symbian dan Android.

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

A. Implementasi Perangkat Lunak

Aplikasi teknik pencarian cerdas menu kuliner ini diimplementasikan untuk sistem operasi Android yang berjalan pada perangkat telepon genggam. Aplikasi ini dirancang menggunakan perangkat lunak atau *software* aplikasi Eclipse Juno dan *software development kit* yang digunakan dalam pemrograman bahasa java. JDK yang digunakan adalah versi 722-windows i586.

B. Pengujian Sistem

Pengujian sistem aplikasi difokuskan pada aplikasi cerdas menu kuliner itu sendiri. Untuk pengujian sistem aplikasi *Libar Cook*, perangkat mobile tidak harus memiliki koneksi dengan internet atau jaringan Wi-Fi ataupun *packet data* dari kartu provider dalam *handphone*, karena aplikasi ini dapat berjalan dengan sistem *offline*. Untuk lebih memperjelas proses dari aplikasi ini, maka pada pengimplementasiannya dibagi berdasarkan fungsi-fungsi dari fitur yang disediakan pada aplikasi ini. Adapun pengujian sistem yang telah dilakukan adalah sebagai berikut :

a. Jendela Halaman Utama Libar Cook

Halaman utama atau welcome screen adalah jendela yang berisi tentang beberapa fasilitas yang dapat diakses oleh *user*, yaitu dengan mengklik *button image* berupa daftar bahan, cari resep, tulis resep dan resep favorit.

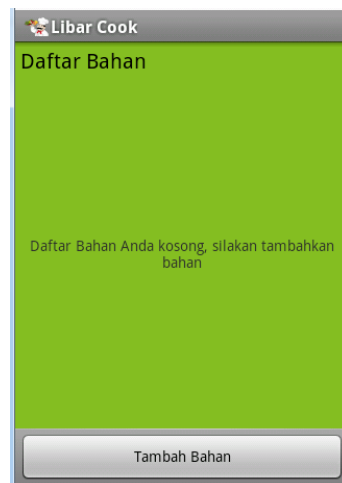


Gambar V.1. Jendela Halaman Utama

Dari gambar diatas, *fitur* daftar bahan terdapat disisi kiri atas dari halaman utama aplikasi. *Fitur* ini berfungsi sebagai tempat penyimpanan bahan pangan. Tepat dibawah daftar bahan terdapat *fitur* tulis resep yang berfungsi sebagai *form* untuk menambahkan resep. Untuk *fitur* cari resep berfungsi untuk melakukan pencarian resep-resep masakan. Dan *fitur* resep favorit sebagai daftar dari resep-resep masakan yang diberikan tanda masakan favorit.

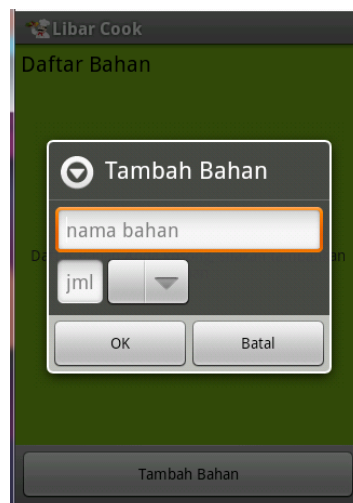
b. Jendela Daftar Bahan

User akan mendapatkan atau menambahkan bahan pangan yang ada pada daftar bahan. Ketika bahan pangan kosong atau *user* ingin menambahkan bahan pangan, maka *user* akan memilih tombol tambah bahan yang tersedia pada aplikasi.



Gambar V.2. Jendela Bahan Pangan

Gambar diatas merupakan tampilan yang muncul ketika *user* memilih *fitur* daftar bahan. Dapat dilihat bahwa isi dari daftar bahan adalah kosong atau dengan kata lain, tidak terdapat bahan pangan. Untuk menambahkan bahan pangan pada daftar bahan, user diharuskan memilih tombol tambah bahan. Berikut adalah tampilan ketika user menambahkan bahan pangan :



Gambar V.3. Jendela Tambah Bahan

Jendela ini akan menampilkan *layout* ketika *user* memilih pilihan daftar bahan kemudian ingin menambahkan bahan pangan yang diinginkan atau bahan pangan yang diperlukan dalam membuat sebuah masakan. Dapat dilihat bahwa terdapat *form* untuk mengisi nama bahan pangan, dan terdapat pula *form* untuk menentukan besar atau jumlah bahan pangan yang ditambahkan beserta dengan satuannya. Pada pengujian kali ini, *user* menambahkan bahan pangan berupa jahe, lengkuas, ayam giling dan bawang merah. Sehingga ketika *user* memilih tombol ok, maka akan tampil sebuah pemberitahuan bahwa bahan pangan berhasil ditambahkan.

Di bawah ini adalah gambar ketika bahan pangan yang diinginkan telah berhasil ditambahkan.



Gambar V.4. Jendela Tambah Bahan Berhasil

c. Jendela Cari Resep

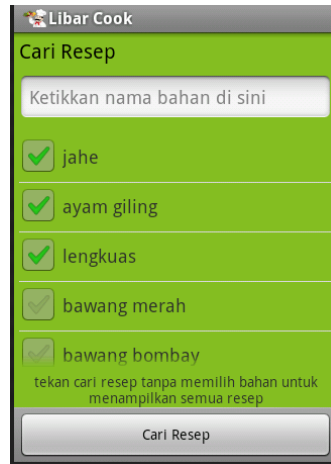
Pada halaman ini, *user* akan dapat melihat beberapa resep masakan yang dapat dikelompokkan berdasarkan jenisnya yaitu minuman panas, minuman dingin, makanan, kue, es krim dan pudding. Berikut adalah tampilan dari halaman atau jendela telusuri resep.



Gambar V.5. Jendela Cari Resep

Gambar diatas adalah hasil pencarian resep secara menyeluruh, sehingga semua resep yang terdapat pada aplikasi ini akan tampil. Resep masakan ini pula dapat dikelompokkan dengan mengklik tombol *down* yang terdapat pada sisi sebelah kanan atas. Sehingga muncul pilihan pengelompokan berdasarkan jenisnya. Aplikasi Libar Cook ini juga dapat menampilkan resep-resep masakan sesuai dengan bahan pangan yang dimiliki atau yang terdapat pada daftar bahan.

Berikut adalah jendela ketika user mencari resep masakan dengan bahan pangan yang ada.



Gambar V.6. Jendela Cari Resep Dengan Bahan Pangan

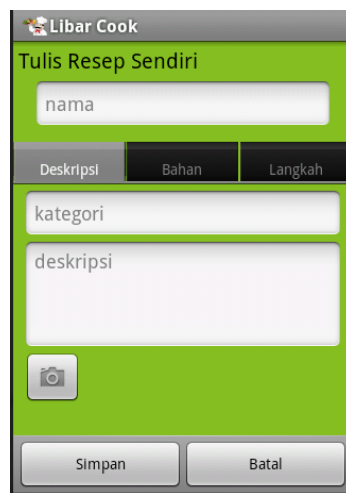
Pada gambar diatas, dapat dilihat bahwa semua bahan pangan yang berada pada kulkas, telah tampil pada *layout* cari resep. Disini *user* dapat memilih bahan pangan yang tersedia dengan cara mencentang bahan pangan tersebut. Ketika memilih pilihan cari resep, maka sistem akan menampilkan resep makanan yang bisa dibuat dengan bahan yang telah dipilih.



Gambar V.7. Jendela Hasil Pencarian Resep Dengan Bahan Pangan

d. Jendela Tulis Resep

Menampilkan halaman berupa deskripsi makanan atau resep yang akan dibuat dan bahan pangan yang dibutuhkan serta langkah membuat sebuah resep. Pada jendela ini juga terdapat tombol kamera yang berfungsi sebagai media pengambil gambar untuk resep masakan yang akan dibuat.



Gambar V.8. Jendela Tulis Resep

Dari tampilan jendela diatas dapat dilihat bahwa untuk *fitur* tulis resep sendiri terdapat *form* yang menjelaskan tentang nama dari resep yang akan dibuat, terdapat pula *form* kategori yang nantinya masakan ini akan dikategorikan sebagai jenis makanan, minuman atau lain sebagainya sesuai dengan jenis makanan yang telah terdapat pada sistem. Terdapat pula *form-form* lainnya seperti bahan, langkah membuat serta *form* gambar dari masakan tersebut.

e. Jendela Resep Favorit

Halaman ini menampilkan semua resep masakan yang menjadi resep favorit dari *user*. Untuk memasukkan sebuah resep pada resep favorit, maka *user* harus memberikan tanda favorit pada resep masakan yang diinginkan untuk menjadi resep favorit.



Gambar V.9. Jendela Resep Favorit

Hasil dari pemilihan resep favorit belum memiliki daftar makanan favorit, dikarenakan *user* yang belum memberikan tanda favorit pada salah satu resep masakan yang terdapat pada *fitur* cari resep.

BAB VI

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari pembahasan sebelumnya dapat disimpulkan :

1. Aplikasi teknik pencarian cerdas menu kuliner ini dapat mengkreasikan ide-ide tentang kuliner. Dan dapat menjadi pemandu dalam hal mencari dan mengetahui resep masakan.
2. Dengan aplikasi ini, persoalan masak memasak dan mengetahui resep-resep masakan dapat dilakukan dimana saja dan kapan saja.
3. Aplikasi ini terdiri dari daftar bahan yang berfungsi sebagai penampung bahan pangan yang dibutuhkan dalam membuat masakan.
4. Aplikasi ini dapat terhubung dengan database bahan pangan untuk pendeteksian bahan makanan secara otomatis.
5. Aplikasi ini dapat diakses pada perangkat *mobile* berbasis Android versi 2.2 (Froyo) sampai 4.2 (Jelly Bean).

B. Saran

Berdasarkan hasil penelitian dan kesimpulan, maka disarankan :

1. Aplikasi ini dapat digunakan semaksimal mungkin dalam hal mengkreasikan ide-ide dan gagasan untuk membuat masakan.
2. Aplikasi ini dapat dikembangkan dengan menambahkan *fitur-fitur* yang kompatible agar dapat berjalan di semua *platform* sistem operasi *mobile*.

DAFTAR PUSTAKA

- Al Fatta, Hanif. *Analisis dan Perancangan Sistem Informasi : untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*, Andi Offset, Yogyakarta, 2007.
- Ambarini., Effendi, Hartati, Yekti, Dr. *Menu Sehari-hari Untuk Golongan Darah A*, Gramedia Pustaka Utama, Jakarta, 2008.
- Badiatul, Muchlisin, Asti., Nurisysyafa'ah, Laela. *Kumpulan Resep Masakan Tradisional Dari Sabang Sampai Marauke*, PT. Buku Kita, Jakarta, 2009.
- Budiharto, Widodo. *Belajar Sendiri : Membuat Robot Cerdas, Panduan Utama Untuk Mengikuti Kontes Robot*, PT. Elex Media Komputindo, Jakarta, 2007.
- Cahana, P., Irwan, Suhandi. *Makan Sehat Hidup Sehat*, PT. Kompas Media Pratama, Jakarta, 2006.
- Eka, Pratiwiningsih. *Aplikasi Restoran Untuk Pemesanan Menu Makanan dan Minuman Berbasis Visual Basic 6.0*, Skripsi, Politeknik Telkom Bandung, Bandung, 2010.
- Fizqi, Gofur, Abdul. *Rancang Bangun Aplikasi Resep Masakan Khas Indonesia Berbasis Mobile Android Pada Kelompok Pkk Di Kelurahan Lebak Gede Bandung*, Skripsi, UNIKOM, Bandung, 2012.
- Gunadi, Hantoro. *Mobile Broadband : Tren Teknologi Saat Ini dan Masa Datang*, Penerbit Informatika, Bandung, 2008.
- Herianto, Tjendry. *Teknik Pemrograman Turbo Prolog Tingkat Lanjut*, Penerbit Andi, Yogyakarta, 1993.
- Hendrayudi, VB 2008 *Untuk Berbagai Keperluan Pemrograman*, PT. Elex Media Komputindo, Jakarta, 2008.
- Joyokusumo, Nuraida. *Warisan Kuliner Keraton Yogyakarta*, Gramedia Pustaka Utama, Jakarta, 2009.

Mulyadi, *Android App Inventor*, Multimedia Center Publishing, Yogyakarta, 2011.

Murphy, Mark, L. *Beginning Android 2*. United States of America: Aapress. 2010.

Pinontoan, M., Soeherman, Bonnie. *Designing Information System : Concepts & Cases with Visio*, PT. Elex Media Komputindo, Jakarta, 2008.

Pressman, S, Roger. *Software Engineering: A Practitioner's Approach*.
Diterjemahkan oleh LN Harnaningrum dengan judul: *Rekayasa Perangkat Lunak : Pendekatan Praktis*. Edisi II. Andi, Yogyakarta, 2002.

Reto, Meier. *Professional Android 2 Application Development*, Indiana : Wiley Publishing. 2010.

Safaat, H, Nazruddin. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Edisi Revisi Penerbit Informatika, Bandung, 2012.

Safaat, H, Nazruddin. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Penerbit Informatika, Bandung, 2011.

Supardi, Yuniar, Ir. *Semua Bisa Menjadi Programmer Android Basic*, PT. Elex Media Komputindo, Jakarta, 2011.

Surya, Suparlan. *Pembangunan Sistem Rekomendasi Menu Makanan Restoran Berbasis Web Menggunakan Metode Knowledge Based Recommendation*, Skripsi, Universitas Pendidikan Indonesia, 2012.

Wahyu, R.A. *Aplikasi Maskulin Untuk Sistem Operasi Android 2.2 Menggunakan Java Eclipse*, Skripsi, Gunadarma, Depok, 2011.

Yayasan Penyelenggara Penerjemah/Penafsir AlQuran, *Al Quran dan Terjemahannya* Departemen Agama, Jakarta, 1971.

<http://cai-sl.blogspot.com/2012/06/pengertian-bahan-pangan-hewani-dan-nabati.html>.
Diakses 18 Juni 2013.

<http://resepmasakanindonesia.info/info-kuliner-sejarah-masakan.html> Diakses 17 Juni
2013

RIWAYAT HIDUP PENULIS



Layli Latifah Lestary akrab dengan panggilan Lely, lahir di Makassar, 17 Oktober 1989. Anak kedua dari tiga bersaudara. Anak dari pasangan Bapak Drs. H. Ansar Djafar. dan Ibu Dra. Hj. Hasniah. ini lebih senang bergaul dengan keluarga dan para sahabat. Aktifitas setelah pulang dari dunia perkuliahan lebih sering di rumah, membaca dan mendengar musik.

Pendidikan yang pernah ditempuh :

1. Sekolah Dasar pada tahun 1995-2001 di SD Inpres Minasa Upa.
2. Sekolah Menengah Pertama pada tahun 2001-2004 di SMP Negeri 21 Makassar.
3. Sekolah Menengah Kejuruan pada tahun 2004-2007 di SMA Negeri 9 Makassar.
4. Melanjutkan pendidikan pada Universitas Islam Negeri (UIN) Alauddin Makassar, fakultas Sains dan Teknologi pada tahun 2007, mengambil jurusan Teknik Informatika.

Hoby :

1. Membaca novel dan komik.
2. Mendengar musik, jalan-jalan dan shopping.